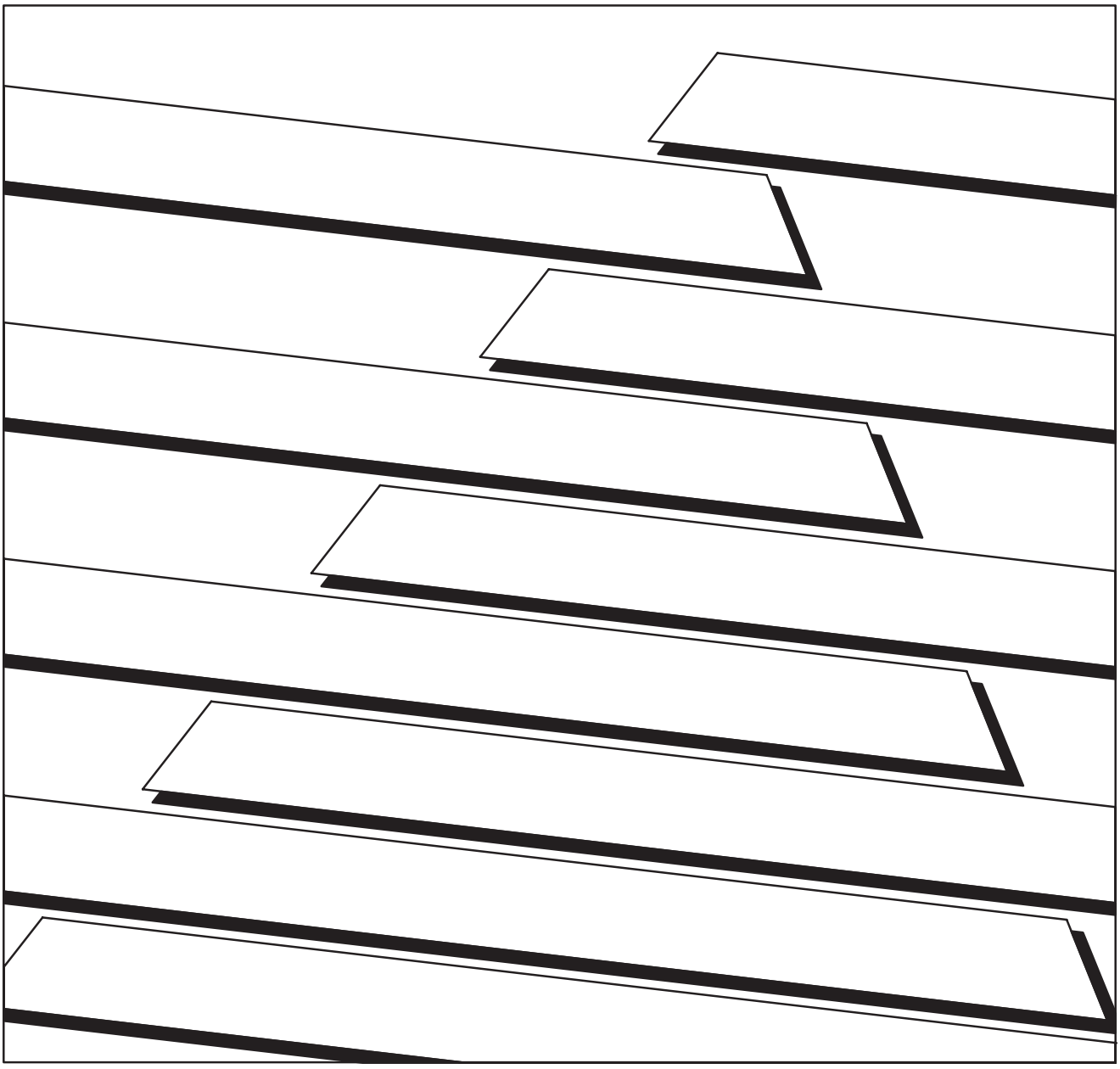
ALLEN-BRADLEY

Linear Positioning Module

Cat. No. 1771-QB

User Manual



Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, the Allen-Bradley Company, Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley Publication SGI-1.1, “Safety Guidelines for the Application, Installation and Maintenance of Solid State Control” (available from your local Allen-Bradley office) describes some important differences between solid-state equipment and electromechanical devices which should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted manual, in whole or in part, without written permission of the Allen-Bradley Company Inc. is prohibited.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attentions help you:

- identify a hazard
- avoid the hazard
- recognize the consequences

Important: Identifies information that is especially important for successful application and understanding of the product.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Table of Contents

Preface **P-1**

Organization of the Manual P-1
Audience P-2
Related Publications P-2
Related Software P-2
Frequently Used Terms P-3

Introducing the Linear Positioning Module **1-1**

What is the Linear Positioning Module? 1-1
Product Compatibility 1-2
Transducers 1-2
Servo and Proportional Valves 1-3
System Overview 1-4

Positioning Concepts **2-1**

Axis Motion 2-1
Closed-Loop Positioning 2-2
Linear Displacement Transducer 2-2
A Simple Positioning Loop 2-3
Proportional Gain 2-4
Feedforwarding 2-5
Integral Control (Reset Control) 2-5
Derivative Control (Rate Control) 2-6
Deadband 2-7
PID Band 2-7

Positioning with the Linear Positioning Module **3-1**

How the Module Fits in a Positioning System 3-1
How the Module Interacts with a PLC 3-2
Read Operations 3-2
Write Operations 3-2
Axis Movement 3-2
Commanding Motion 3-4
Setpoints 3-4
Jogging 3-5
Motion Blocks 3-5

Hardware Description 4-1

Indicators 4-1
Wiring Arm Terminals 4-2
Transducer Interface 4-3
Determining the Optimum Number of Circulations 4-3
Discrete Inputs 4-5
Auto/Manual Input 4-6
Hardware Start Input 4-6
Hardware Stop Input 4-7
Jog Forward Input 4-7
Jog Reverse Input 4-7
General Purpose Inputs 4-7
Analog Output Interface 4-7
Discrete Outputs 4-8
OUTPUT 1 4-9
OUTPUT 2 4-9
Power Supplies 4-9

Installing the Linear Positioning Module 5-1

Before You Begin 5-1
Avoiding Backplane Power Supply Overload 5-1
Planning Module Location 5-1
Electrostatic Discharge 5-2
Setting Analog Output Switches 5-2
Keying 5-5
Inserting the Module 5-5
Wiring Guidelines 5-6
Using Shielded Cables 5-6
Using Twisted Wire Pairs 5-8
Connecting AC Power 5-8
Power Supplies 5-10
Connecting the Transducer Interface 5-10
Power Supply 5-11
Transducer Interface 5-12
Connecting the Discrete Inputs 5-12
Power Supply 5-14
Auto/Manual Input 5-14
Hardware Start Input 5-14
Hardware Stop Input 5-14
Jog Forward Input 5-15
Jog Reverse Input 5-16
General Purpose Inputs 5-16
Connecting Multiple Modules 5-16

Connecting the Analog Outputs 5-18
Power Supply 5-19
Analog Output 5-19
Connecting the Discrete Outputs 5-20
Power Supply 5-21
OUTPUT 1 5-21
OUTPUT 2 5-21

Interpreting Module-to-PLC Data (READS) 6-1

PLC Communication Overview 6-1
Status Block 6-1
Word Assignment 6-2
Module Configuration Word (word 1) 6-2
Status Word 1 (words 2 and 6) 6-3
Status Word 2 (words 3 and 7) 6-7
Position/Error/Diagnostic Words 6-9
Active Motion Segment/Setpoint (words 10 and 11) 6-13
Measured Velocity (words 20 and 21) 6-14
Desired Velocity (words 22 and 23) 6-14
Desired Acceleration (words 24 and 25) 6-15
Desired Deceleration (words 26 and 27) 6-15
Percent Analog Output (words 28 and 29) 6-16
Maximum Velocity (words 30, 31 and 32, 33) 6-17

Formatting Module Data (WRITES) 7-1

Data Blocks Used in Write Operations 7-1
Parameter Block (Required) 7-1
Setpoint Block (Optional) 7-1
Command Block (Required) 7-1
Parameter Block 7-1
Parameter Control Word (word 1) 7-3
Analog Range (words 2 and 31) 7-5
Analog Calibration Constants (words 3, 4 and 32, 33) 7-6
Transducer Calibration Constant (words 5, 6 and 34, 35) 7-7
Zero-Position Offset (words 7, 8 and 36, 37) 7-8
Software Travel Limits (words 9, 10 and 38, 39) 7-9
Zero-Position and Software Travel Limit Examples 7-10
In-Position Band (words 11 and 40) 7-13
PID Band (words 12 and 41) 7-14
Deadband (words 13 and 42) 7-15
Excess Following Error (words 14 and 43) 7-16
Maximum PID Error (words 15 and 44) 7-16
Integral Term Limit (words 16 and 45) 7-17
Proportional Gain (words 17 and 46) 7-18
Gain Break Speed (words 18 and 47) 7-19

Gain Factor (words 19 and 48) 7-20
Integral Gain (words 20 and 49) 7-21
Derivative Gain (words 21 and 50) 7-22
Feedforward Gain (words 22 and 51) 7-22
Global Velocity (words 23 and 52) 7-23
Global Acceleration/Deceleration (words 24, 25 and 53, 54) 7-24
Velocity Smoothing (Jerk) Constant (words 26 and 55) 7-24
Jog Rate (Low and High) (words 27, 28 and 56, 57) 7-26
Reserved (words 29, 30 and 58, 59) 7-27
Setpoint Block 7-27
Setpoint Block Control Word (word 1) 7-29
Incremental/Absolute Word (word 2) 7-29
Setpoint Position 7-30
Local Velocity 7-31
Local Acceleration/Deceleration 7-32
Command Block 7-32
Axis Control Word 1 (words 1 and 8) 7-33
Axis Control Word 2 (words 2 and 9) 7-38
Setpoint 13 Words (words 3 to 7 and 10 to 14) 7-39

Initializing and Tuning the Axes 8-1

Before You Begin 8-1
Adjusting the Servo Valve Nulls 8-2
Initializing the Parameter Block 8-2
Verifying Analog Output Polarity 8-7
Verifying Transducer Calibration Constants 8-7
Axis Tuning 8-10
Analog Calibration Constants 8-10
Feedforward Gain 8-11
PID Loop Gains 8-12
Update the Application Program 8-13

Advanced Features 9-1

Motion Block 9-1
Motion Block Control Word 9-4
Programmable Input and Output 9-5
Programmable I/O Control Word 9-5
Default I/O Configuration 9-8
Motion Segments 9-8
Motion Segment Control Words 9-8
Desired Position, Local Velocity, Local Acceleration and Local Deceleration Words 9-11
Trigger Velocity/Position Words 9-11
The Command Block and the Motion Block 9-11
The Status Block and the Motion Block 9-11

Using the Motion Block 9-12

Sample Application Programs 10-1

Programming Objectives 10-1
Block Transfer Sequencing 10-2
PLC-5 Block Transfer Instructions 10-3
Application Program #1 10-3
Planning the Data Blocks for Application Program #1 10-5
Program Rungs for Application Program #1 10-8
Application Program #2 10-11
Planning the Data Blocks for Application Program #2 10-12
Program Rungs for Application Program #2 10-16

Troubleshooting 11-1

Fault Indicators 11-1
Module Fault Indicator 11-2
Loop Active Indicators 11-2
Indicator Troubleshooting Guide 11-2
Troubleshooting Feedback Faults 11-3
Troubleshooting Flowchart 11-4
Flowchart Notes 11-7

Glossary of Terms & Abbreviations A-1

Status Block B-1

Parameter Block C-1

Setpoint Block D-1

Command Block E-1

Motion Block F-1

Hexadecimal Data Table Forms G-1

Data Formats H-1
BCD H-1
2's Complement Binary H-1
Bit Inversion Method H-2
Subtraction Method H-2
Implied Decimal H-2
Position Format H-3
Double Word Position Format H-4

Product Specifications I-1

Preface

This manual explains how to install and configure the Linear Positioning Module. It includes sample application programs to illustrate how to program a PLC™ to work with the Linear Positioning Module.

Organization of the Manual

This manual contains eleven chapters and nine appendices that address the following topics:

Chapter	Title	Describes:
1	Introducing the Linear Positioning Module	the functions and features of the Linear Positioning Module
2	Positioning Concepts	concepts and principles of closed-loop servo positioning
3	Positioning with the Linear Positioning Module	using the Linear Positioning Module in a positioning system
4	Hardware Description	module hardware, module interfaces, and other hardware items you need for a positioning system
5	Installing the Linear Positioning Module	configuring the module's analog outputs and installing the module in your system
6	Interpreting Module-to-PLC Data (READS)	monitoring module operation from a logic controller by reading and interpreting data that the module transfers to the logic controller's data tables
7	Formatting Module Data (WRITES)	formatting parameter, move description, and control data for block transfers to the Linear Positioning Module
8	Initializing and Tuning the Axes	bringing the module online
9	Advanced Features	using the motion block to perform blended moves; using programmable input and output operations
10	Sample Application Programs	two application programs, one using basic concepts and the other using advanced features, to control and monitor the module
11	Troubleshooting	using the module's indicators and the status block to diagnose and remedy module faults and errors
Appendix A	Glossary	common terms and abbreviations
Appendix B	Status Block	status block word assignments
Appendix C	Parameter Block	parameter block word assignments
Appendix D	Setpoint Block	setpoint block word assignments

Chapter	Title	Describes:
Appendix E	Command Block	command block word assignments
Appendix F	Motion Block	motion block word assignments
Appendix G	Hexadecimal Data Table Form	hexadecimal data worksheets
Appendix H	Data Formats	valid data formats
Appendix I	Product Specifications	1771-QB product specifications

Audience

Read this manual if you intend to install or use the Linear Positioning Module (Cat. No. 1771-QB).

To use the module, you must be able to program and operate an Allen-Bradley PLC. In particular you must be able to program block transfer instructions.

In this manual, we assume that you know how to do this. If you don't, refer to the *User Manual* for the PLC you'll be programming.

Related Publications

Consult the Allen-Bradley Industrial Computer Division Publication Index (SD 499) if you would like more information about your modules or PLCs. This index lists all available publications for Allen-Bradley programmable controller products.

Related Software

The Hydraulics Configuration and Operation Option (Cat. No. 6190-HCO) operates within the ControlView Core (Cat. No. 6190-CVC) environment to provide full configuration and realtime monitoring for the Linear Positioning Module. Both software packages are available from:

Allen-Bradley Company, Inc.
1201 South Second Street
Milwaukee, WI 53204
(414) 382-2000

Servo Analyzer is a software package that aids in tuning the axes by letting you display an axis profile as you tune it. The resulting graphics may be plotted, printed or saved to a file. The software is available from:

Computer Software Design
P.O. Box 962
Roseburg, OR 97470
(503) 673-8583

Frequently Used Terms

Appendix A contains a complete glossary of terms and abbreviations used in this manual.

To make this manual easier for you to read and understand, product names are avoided where possible. The Linear Positioning Module is also referred to as the “module”.

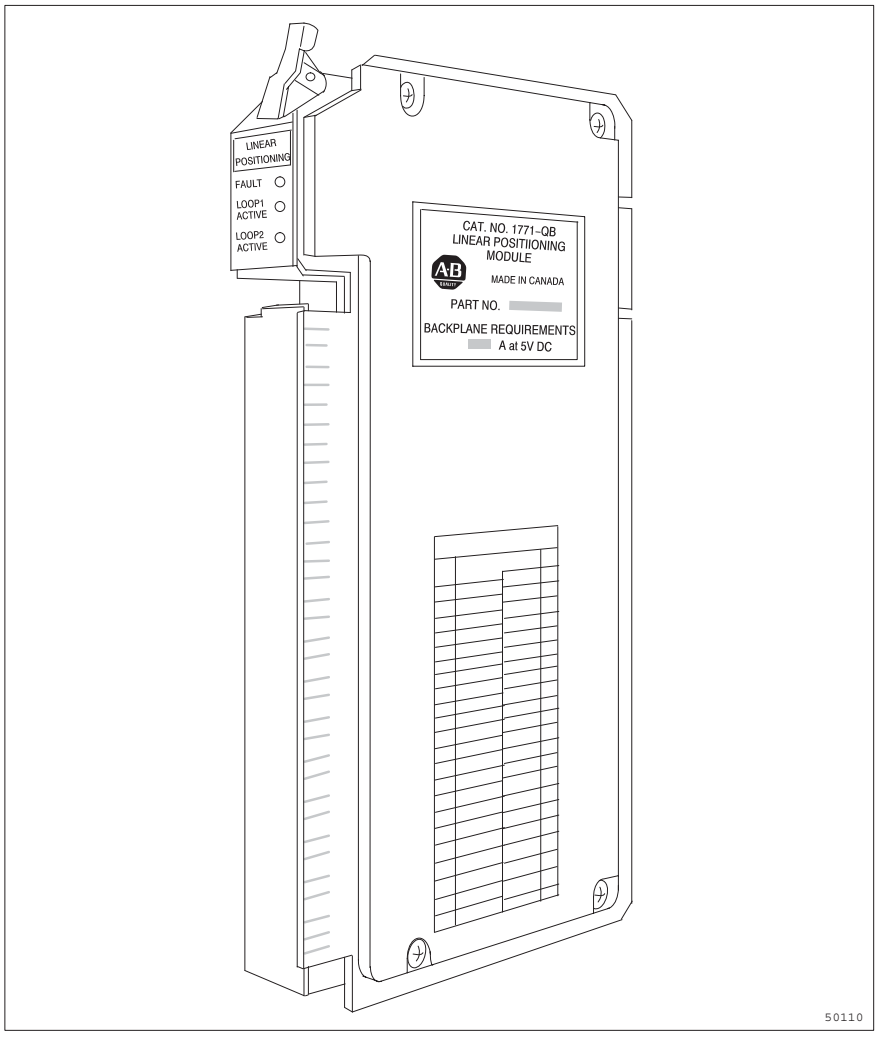
Chapter 1

Introducing the Linear Positioning Module

What is the Linear Positioning Module?

The Linear Positioning Module (Cat. No. 1771-QB) is a dual-loop position controller occupying a single slot in the Allen-Bradley 1771 Universal I/O chassis. It can control servo or proportional hydraulic valves, or some electric servos. Position is measured with a linear displacement transducer. You use the module to control and monitor the linear position of a tool or workpiece along one or two axes.

Figure 1.1
Linear Positioning Module



Product Compatibility

PLCs

You can use the module with any Allen-Bradley PLC that uses block transfer programming in local 1771 I/O systems including:

- PLC-2 family
- PLC-3 family
- PLC-5 family
  - PLC-5/10 (Cat. No. 1785-LT4)
  - PLC-5/11 (Cat. No. 1785-LT11)
  - PLC-5/12 (Cat. No. 1785-LT3)
  - PLC-5/15 (Cat. No. 1785-LT)
  - PLC-5/20 (Cat. No. 1785-L20)
  - PLC-5/25 (Cat. No. 1785-LT2)
  - PLC-5/30 (Cat. No. 1785-L30)
  - PLC-5/40 (Cat. No. 1785-L40)
  - PLC-5/60 (Cat. No. 1785-L60)

Transducers

The Linear Positioning (QB) Module is compatible with linear displacement transducers manufactured by:

MTS Systems Corporation
Sensors Divisions
Box 13218, Research Triangle Park
North Carolina 27709
(919) 677–0100

Balluff Inc.
P.O. Box 937
8125 Holton Drive
Florence, KY 41042
(606) 727–2200

Santest Co. Ltd.
c/o Ellis Power Systems
123 Drisler Avenue
White Plains, NY 10607
(914) 592-5577

Lucas Schaevitz Inc.
7905 N. Route 130
Pennsauken, NJ 08110-1489
(609) 662-8000

All four manufacturers provide versions of the transducer that connect directly to the module's wiring arm, without an external digital interface box. The module may also be compatible with other linear displacement transducers.

Servo and Proportional Valves

The module provides current ranges of up to ±100 mA for direct interface to most servo valves, most proportional valves, and a ±10 volt option for compatibility with other devices, such as electric servo interfaces. The module is compatible with valves supplied by the following manufacturers:

- Moog, East Aurora NY — servo/proportional
- Parker Hannifin Corporation, Elyria OH — servo/proportional
- Robert Bosch Corporation — proportional
- Rexroth Corporation, Lehigh Valley PA — servo/proportional
- ATOS — proportional
- Atchley, Canaga Park CA — servo
- Pegasus — servo
- Vickers Inc., Grand Blanc, MI — servo/proportional

The module may also be compatible with other valves.

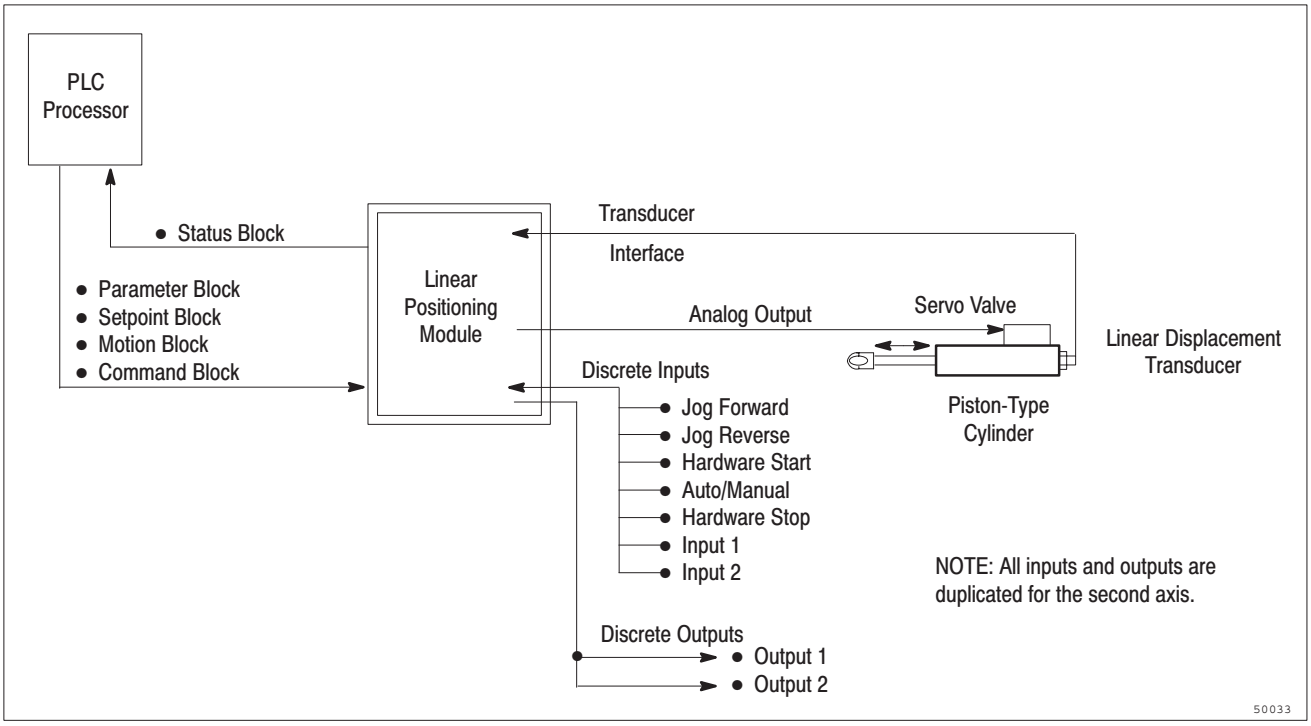
Important: Some proportional valves with LVDT loop controllers may limit the module's output and thus prevent the module from providing optimal control.

System Overview

Figure 1.2 shows one of the module's two control loops within a linear positioning system for closed-loop axis control. The module communicates with a programmable controller through the 1771 backplane.

The programmable logic controller sends commands and user-programmed data from the data table to the module as directed by a block-transfer write instruction.

Figure 1.2
System Overview



Using PLC programming, you can:

- send configuration and control parameters to the module via parameter, setpoint, motion, and command blocks. With this data the module determines axis parameters, calculates velocity curves, and commands axis end-positions. (See Chapters 7 and 9.)
- read status blocks to monitor axis position and status indicators in your process control system. (See Chapter 6.)

The module's analog outputs (one for each control loop) connect to servo or proportional valves via wiring arm terminals. The module controls speed and position by adjusting the voltage or current levels of the analog outputs 500 times each second.

The module also connects to linear displacement transducers (one for each of the two axes) via wiring arm terminals. The transducer senses the axis position and feeds it back to the module, thereby closing the control loop.

The module's built-in processor samples the linear displacement transducer interfaces and determines positions along each of the two axes *every two milliseconds*. The module then updates the analog outputs based on a proprietary algorithm designed specifically to handle hydraulic actuators. This rapid update rate provides repeatable positioning and superior control of velocity without jerky movement.

Motion blocks provide for complex motions by allowing motion segments to be blended or chained together. These motion segments may also be synchronized using the hardware input triggers and outputs.

Cam emulation permits motion segments in one axis to start motion segments in another axis. Articulated motions and axis sequencing may be easily accomplished.

Chapter 2

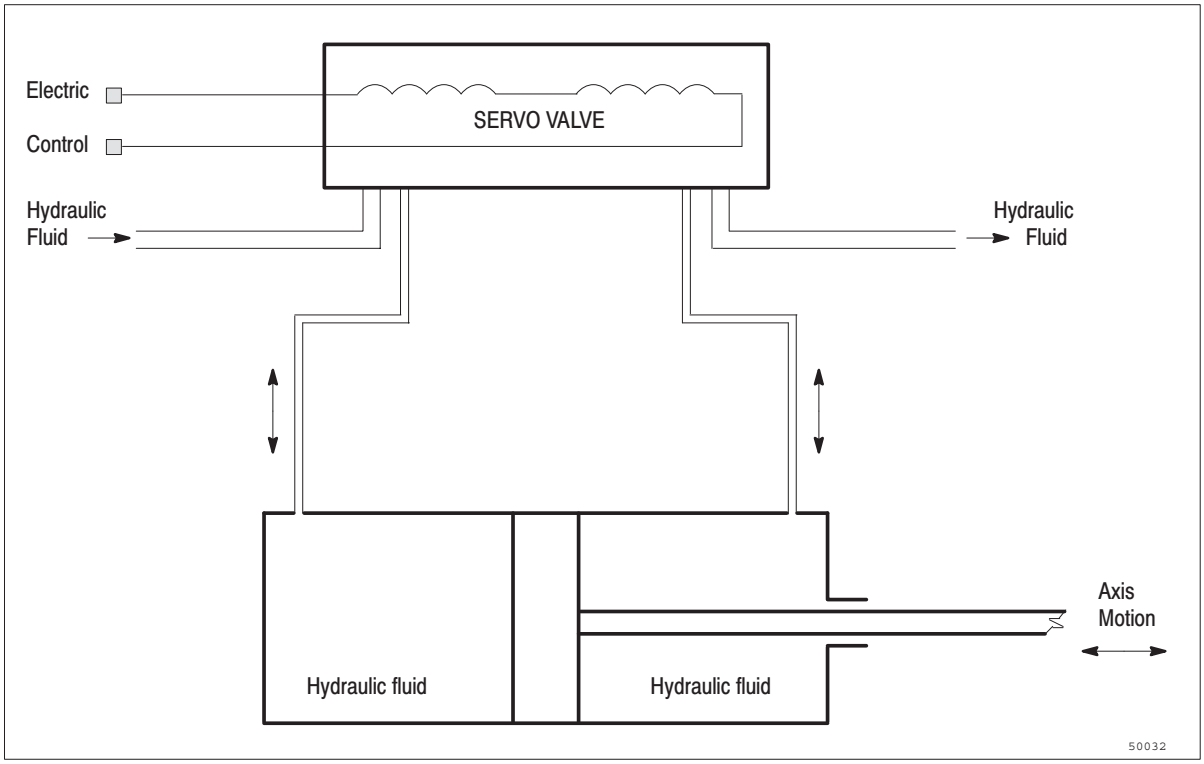
Positioning Concepts

This chapter explains concepts and principles of axis positioning. If you are thoroughly familiar with the concepts of closed-loop servo positioning, you can go on to Chapter 3.

Axis Motion

Figure 2.1 illustrates a typical method of converting the flow of fluid into a linear displacement.

Figure 2.1
Piston-Type Hydraulic Cylinder



The servo valve controls the flow of hydraulic fluid into or out of the hydraulic cylinder. Adding fluid to the left side of the cylinder extends the rod; adding fluid to the right side retracts it.

Closed-Loop Positioning

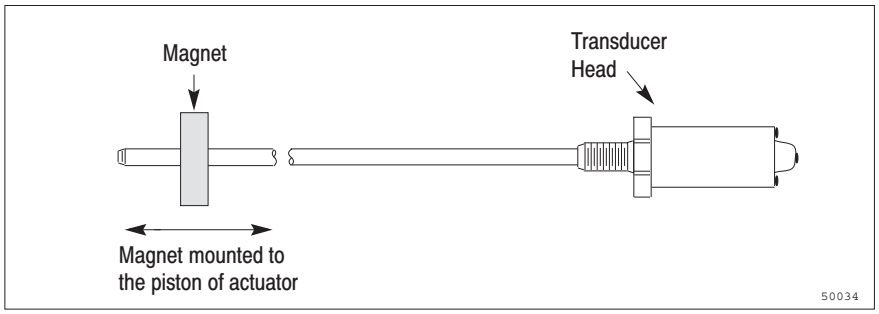
Closed-loop positioning is a precise means of moving an object from one position to another. In a typical application, a positioning device activates a servo valve controlling the movement of fluid in a hydraulic system. The movement of fluid translates into the linear motion of a hydraulic cylinder. A transducer monitors this motion and feeds it back to the positioning device. The positioning device, in turn, calculates a positioning correction and feeds it back to the servo valve.

Important: Throughout this manual we refer to servo valves, but you can also use the analog outputs to control proportional valves or an electric servo.

Linear Displacement Transducer

A linear displacement transducer (see Figure 2.2) is a device that senses the position of an external magnet to measure displacements.

Figure 2.2
Linear Displacement Transducer

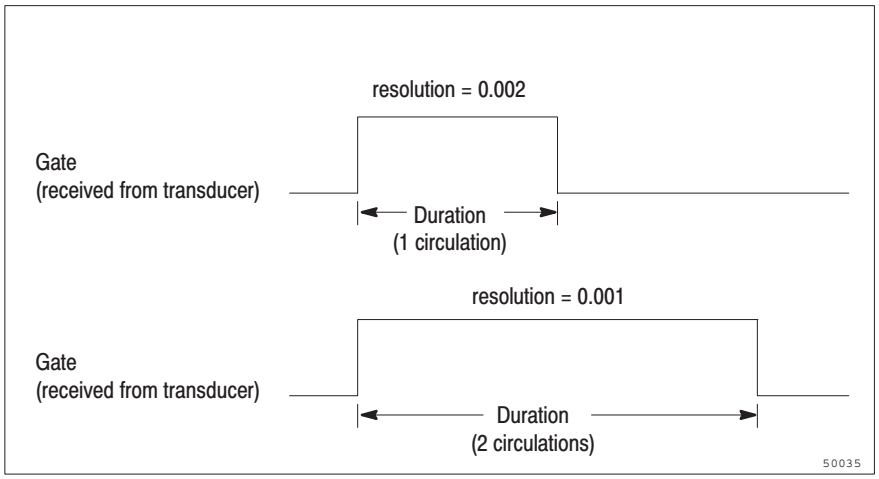


The transducer sends a signal through the transducer wave guide where a permanent magnet generates the return pulse. You can use the time interval between the transducer's signal and the return pulse to measure axis displacement.

Circulations

Some linear displacement transducers provide circulations or recirculation to improve resolution. (See Figure 2.3.) This technique stretches the pulse by a factor of two or more and results in finer resolution in the circuitry monitoring the pulse width.

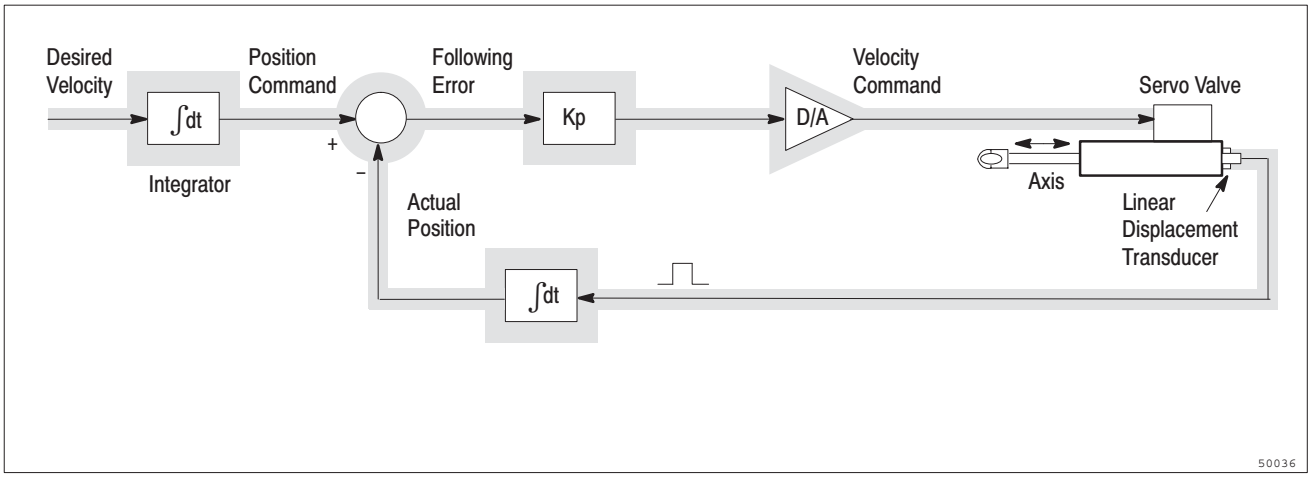
Figure 2.3
Circulations



A Simple Positioning Loop

To move a specified distance along an axis, you can command the hydraulic device to move at a specific velocity for a specific length of time. However, this method can be imprecise. To control the position of the hydraulic device accurately you need a loop to monitor actual position. Figure 2.4 shows a simple positioning loop.

Figure 2.4
Positioning Loop



In Figure 2.4:

- **desired velocity** is the desired speed of axis motion from one position to another
- **position command** equals the integration of velocity over time
- **actual position value (transducer feedback)** is the actual position of the axis as measured by the LDT
- **following error** equals position command minus actual position
- **velocity command** is generated by amplifying the following error and converting the result into an analog output
- **D/A** (Digital to Analog convertor) generates the analog output controlling the servo valve
- **K_P** (proportional gain) is the component that causes an output signal to change as a direct ratio of the error signal variation

Proportional Gain

The following error is a function of the velocity command divided by the proportional gain (K_P). To generate the velocity command, multiply the following error by the proportional gain. Proportional gain can be expressed in ips/mil (where 1 mil = 0.001 inches) or mmps/mil (where 1 mil = 0.001 mm).

For example, with a velocity of 12 ips and a gain of 1 ips/mil, the following error is:

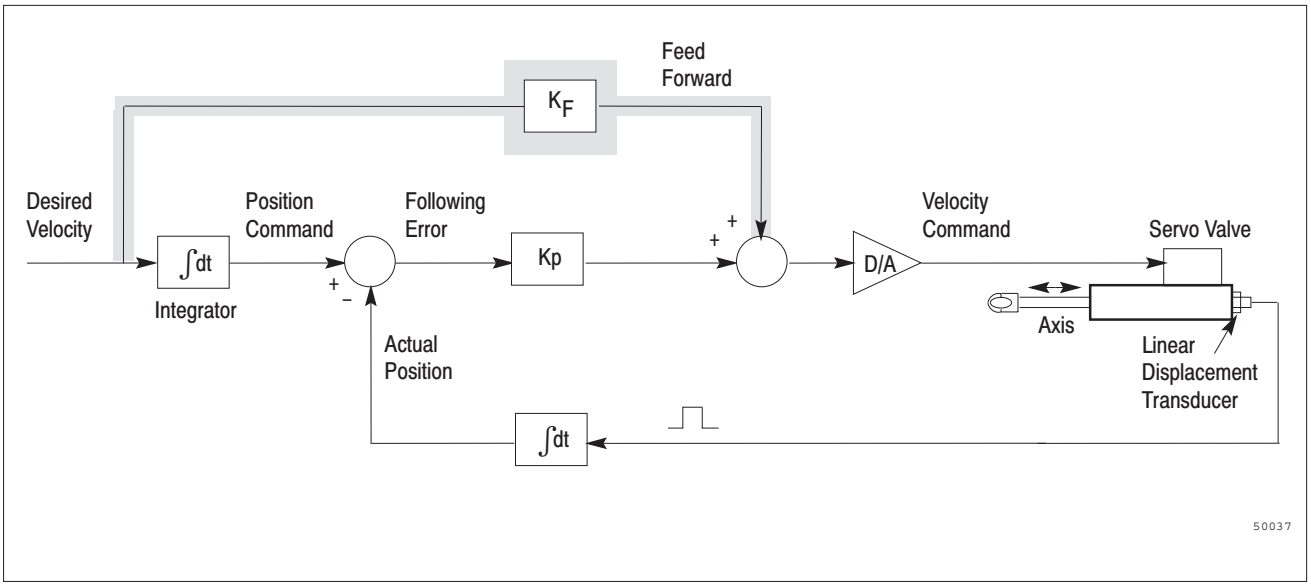
Following Error = Velocity/Gain
= 12 ips/(1 ips/mil)
= 12 mil

When you increase the gain, you decrease the following error and decrease the cycle time of the system. However, the capabilities of the system limit the gain. Too large a gain causes instability.

Feedforwarding

To decrease the following error without increasing the gain, you can add a feedforward component. (See Figure 2.5.)

Figure 2.5
Positioning Loop with Feedforwarding



Feedforwarding requires an additional summing point and an amplifier. Multiply the desired velocity by the feedforward gain K_F to produce a feedforward value. The feedforward value, added to a multiplication of the following error by the proportional gain (K_P), generates the velocity command.

Without feedforwarding, axis motion does not begin until the following error is large enough to overcome friction and inertia. The feedforward component generates a velocity command to move the cylinder almost immediately. This immediate response keeps the actual position closer to the desired position and thereby reduces the following error.

Integral Control (Reset Control)

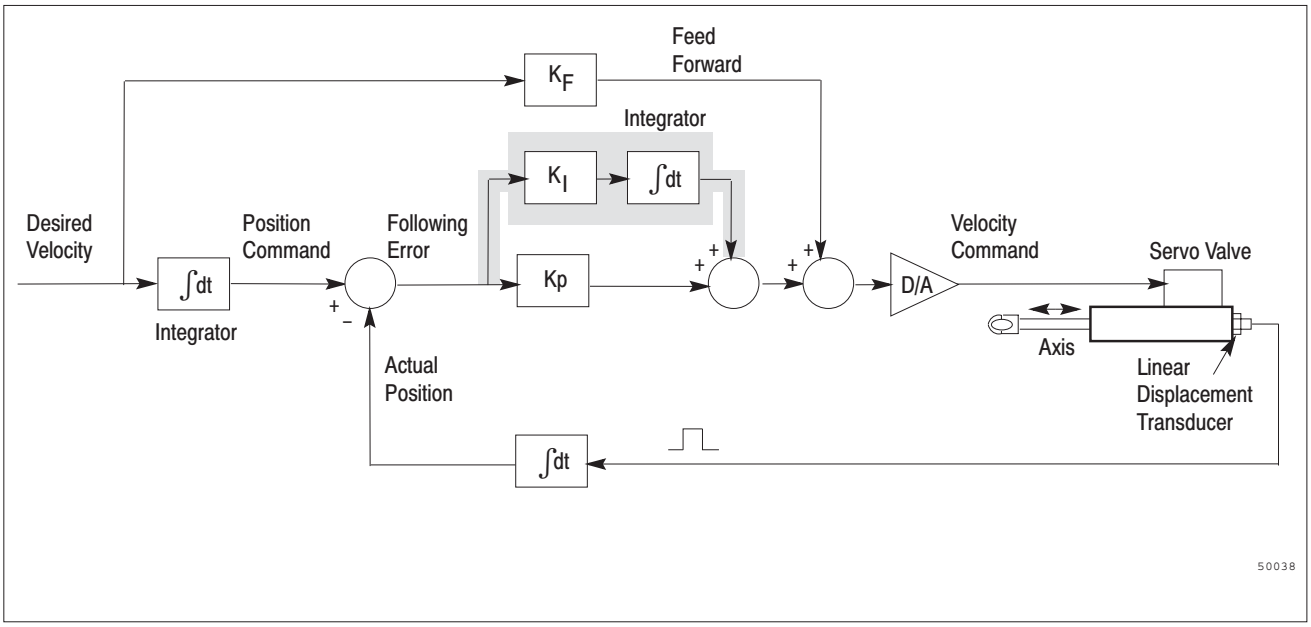
You can increase the positioning accuracy of the control loop by adding an integral component. (See Figure 2.6.)

To achieve the integral component of the positioning loop, integrate the following error over time and amplify it to produce an integral value. Then add this integral value to the proportional component and the feedforward value to generate the velocity command.

Without integral control, the axis responds only to the size of the positioning error, not its duration. Integral control responds to both the size and duration of the positioning error. Thus, the integral term continues to adjust the velocity command until it achieves an exact correction.

When you increase the integral gain (K_I), you increase the rate at which the positioning loop responds to a following error. However, the capabilities of the system limit gain K_I. Too large a gain causes instability.

Figure 2.6
Integral Control

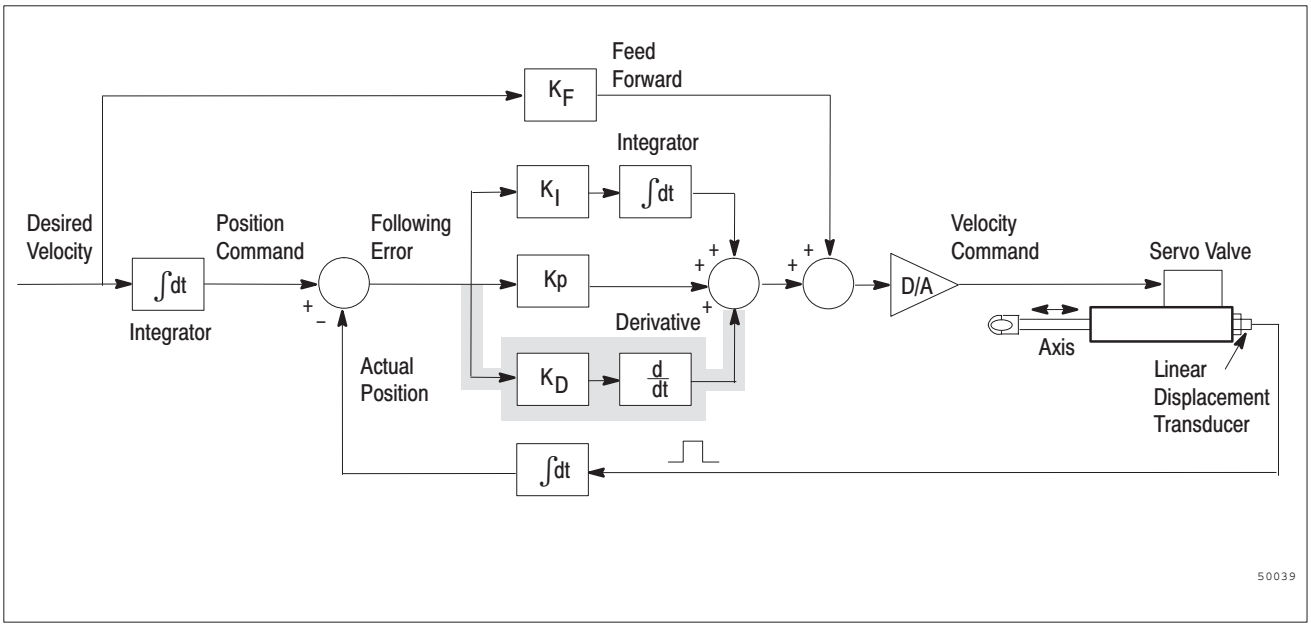


Derivative Control (Rate Control)

Proportional and integral gains can cause instability in a positioning loop. The cylinder can overshoot its programmed endpoints and oscillate or hunt around them. You can increase the stability of the positioning loop by adding a derivative component. (See Figure 2.7.)

Derivative control operates on the rate of change of positioning error. It helps to stabilize the system by opposing changes in positioning error. However, a derivative gain that is too large can cause instability. Derivative control is also very susceptible to electrical noise.

Figure 2.7
Derivative Control



Deadband

Most systems have friction and play in their mechanical linkages. These characteristics can cause a cylinder to oscillate around a programmed endpoint–especially if you use an integral term. You can use a deadband to reduce these oscillations.

A deadband is an area surrounding the programmed endpoint where the error is ignored. Outside the deadband, error is reduced by one half the width of the deadband.

If you apply a deadband to an integral term, the integral output remains constant while the axis is within the deadband. This reduces oscillations around the endpoint. However, if the deadband is too large, it can also reduce the positioning accuracy of the system.

PID Band

Integral and derivative control can cause undesirable results when the axis moves from one position to another. The integral term can cause the axis to overshoot the programmed endpoint. The derivative term opposes changes in error, and thereby changes in position.

You can control the integral and derivative components by defining a PID (proportional, integral and derivative) band. The PID band is a region surrounding the programmed endpoint where the system enables integral or derivative terms. As a result, the integral and derivative components affect only the final positioning of the axis.

Chapter 3

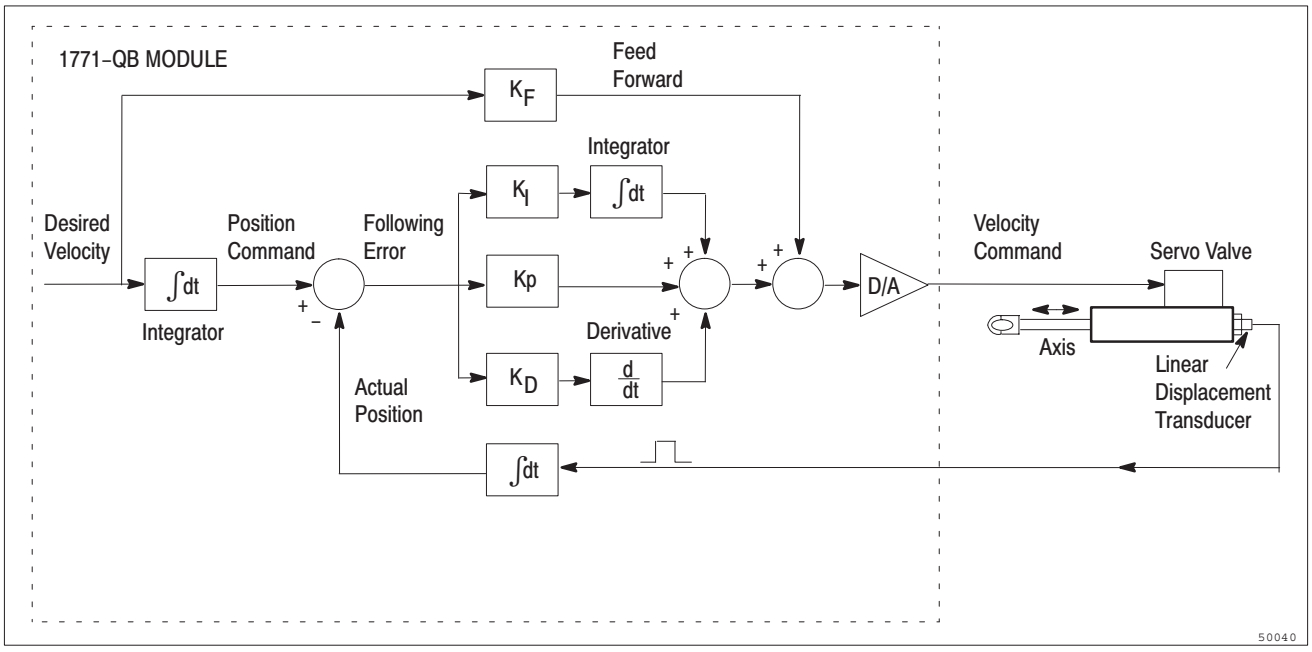
Positioning with the Linear Positioning Module

This chapter explains how the Linear Positioning Module interacts with a programmable controller to control axis movement within a linear positioning system.

How the Module Fits in a Positioning System

Figure 3.1 shows how the module functions in a typical positioning system. Note that the positioning loop closes in the module and functions independently of the programmable controller's I/O scan rate. The fast loop update time of 2 ms is possible, because the module has a built-in microprocessor.

Figure 3.1
The Module in a Positioning System



How the Module Interacts with a PLC

The module is a dual-loop position controller, occupying a single slot in the Allen-Bradley 1771 universal I/O chassis. The module communicates with the PLC through the 1771 backplane. There are two kinds of transfers–read operations and write operations. By programming the PLC you can transfer parameter, setpoint, motion and command blocks to the module to control the two axes. You can also use the PLC to monitor the status of the module's two loops through block read operations. For more details on block transfers, see Chapters 6 and 7.

Read Operations

Read operations enable the programmable logic controller to monitor the status of both axes through the status block. The status block includes detailed information on the two axes: fault conditions, current axis position, positioning error, and diagnostic information.

Write Operations

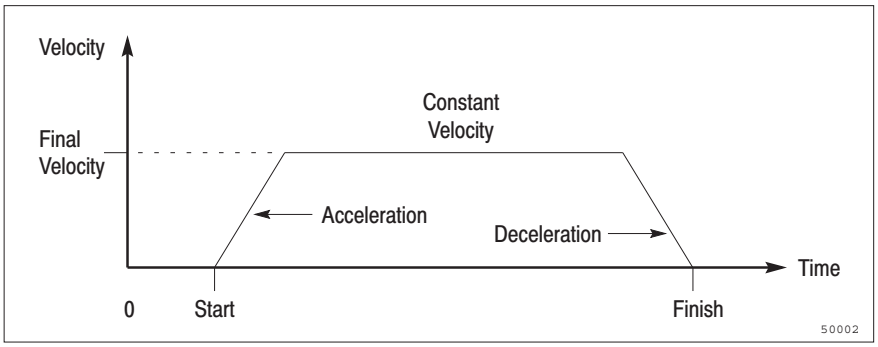
The following four types of write operations enable the programmable controller to control axis movement:

- **Parameter Block** - defines the module's operating parameters for each axis. These parameters include calibration constants, software travel limits, zero-position offset, in-position and PID bands, PID gains, maximum velocities, jog rates, maximum accelerations and decelerations and more.
- **Setpoint Block** - defines up to 12 setpoints for each axis with optional acceleration, deceleration and velocity parameters for each setpoint move. The programmable controller selects from among the 12 setpoints using the command block.
- **Motion Block** - permits complex profiles to be executed by the module. This advanced feature can be used to blend or chain multiple motion segments in a single, continuous motion.
- **Command Block** - you use the command block to select the next setpoint or motion segment to which the axis will move; to set a delayed start, software stop or reset; to set jog bits; to select jog rate (low or high); to set auto/manual, to enable/disable integral control and to define a 13th setpoint.

Axis Movement

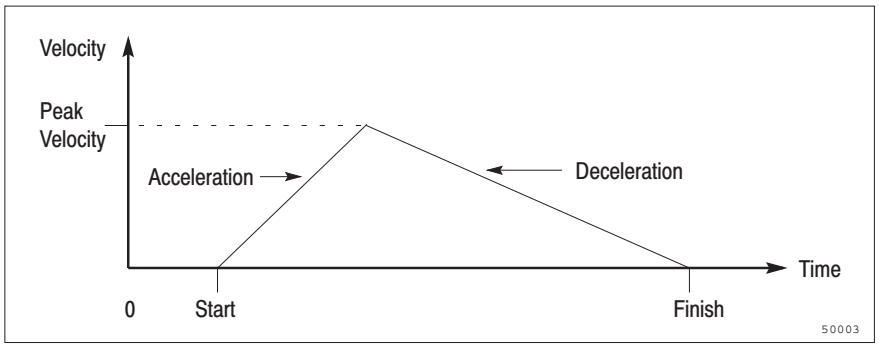
When the module receives a setpoint command, motion segment command, jog command, or a discrete jog input, it automatically calculates the velocity curve for the requested axis movement using parameters that you define for the move. (See Figure 3.2.)

Figure 3.2
Trapezoidal Axis Movement



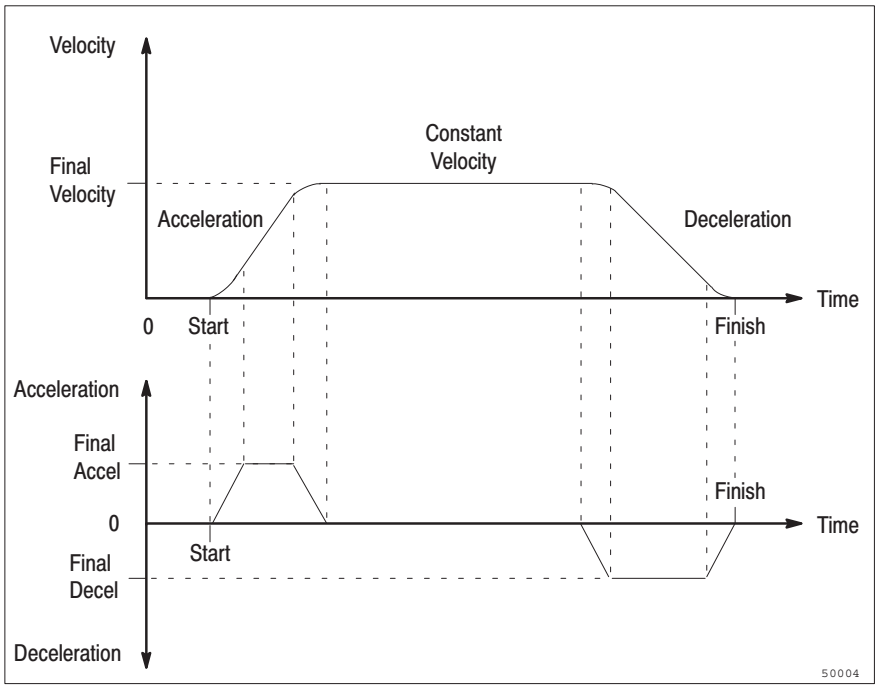
The actuator may not reach the final velocity during a short move which may only consist of acceleration and deceleration phases without a constant velocity phase. This produces a ramp movement. (See Figure 3.3.)

Figure 3.3
Ramp Movement



The module employs a technique called velocity curve smoothing to shape the velocity curve into an “S curve”. To achieve this smoothing, acceleration and deceleration rates are changed to provide more gradual application and removal of force, thus reducing mechanical wear. The velocity smoothing constant that you set in the parameter block determines how quickly acceleration and deceleration change. The lower the value of the velocity smoothing constant, the more slowly acceleration and deceleration change, producing a smoother transition. Figure 3.4 shows the effect of velocity curve smoothing on the axis movement.

Figure 3.4
Axis Movement with Velocity Curve Smoothing



Commanding Motion

There are three ways to specify module axis motion: by setpoints, by jogging or by motion blocks. All motion must be started using the command block and/or hardware inputs.

Setpoints

The module must have the axis controller in auto mode if you are using setpoint moves. You can switch between modes using the auto/manual bit in the command block or the auto/manual discrete input.

Important: The auto/manual bit and the auto/manual input must both be high to enter auto mode.

In the auto mode, you position the actuator by commanding desired setpoints using the command block. You can:

- define up to 12 setpoints through the setpoint block. You can define the 13th setpoint within the command block.
- specify acceleration, deceleration, and velocity for each setpoint move.

- turn on a hardware start enable bit (using the command block), which causes the module to delay movement to the commanded setpoint. The delay ends and movement starts when you activate the hardware start input or send a software start command in the command block.

- command a setpoint while the axis is moving towards another setpoint. If the new setpoint is in the opposite direction of travel, the axis decelerates to zero speed (at the current deceleration rate) and then moves in the opposite direction. If the new setpoint is in the same direction of travel, the old setpoint is abandoned and the axis movement accelerates or decelerates to the specified velocity and continues toward the new setpoint.

Jogging

In the manual mode, you position the actuator by jogging, i.e., directly commanding movement in one direction or the other. You make these movement commands by turning on forward or reverse jog bits (via the command block) or activating forward or reverse hardware jog inputs (typically via momentary action switches).

If you command a jog, the axis movement continues until the actuator reaches the software travel limit or until you turn off the jog bit or jog input, whichever occurs first.

Jog Rates

You define two jog rates (high and low) through the parameter block. You select between low and high jog rates through the jog rate select bit in the command block.

If you change jog rates (from high to low or from low to high) during a jog movement, the axis decelerates/accelerates to the new rate.

Important: Jog commands are ignored in auto mode.

Motion Blocks

A motion block contains information similar to that which the setpoint block uses to define axis movement. In addition, a motion block also contains trigger conditions that will initiate a subsequent axis movement, thus changing the motion of the axis without the intervention of the programmable controller. See Chapter 9 for a full explanation of motion blocks.

Chapter 4

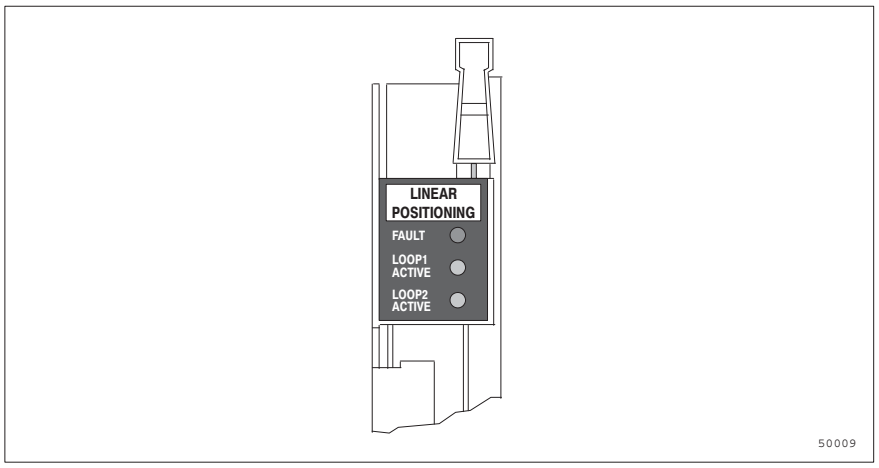
Hardware Description

This chapter describes the Linear Positioning Module hardware, as well as other hardware required for a positioning system.

Indicators

Figure 4.1 shows the three indicators on the module.

Figure 4.1
Indicators



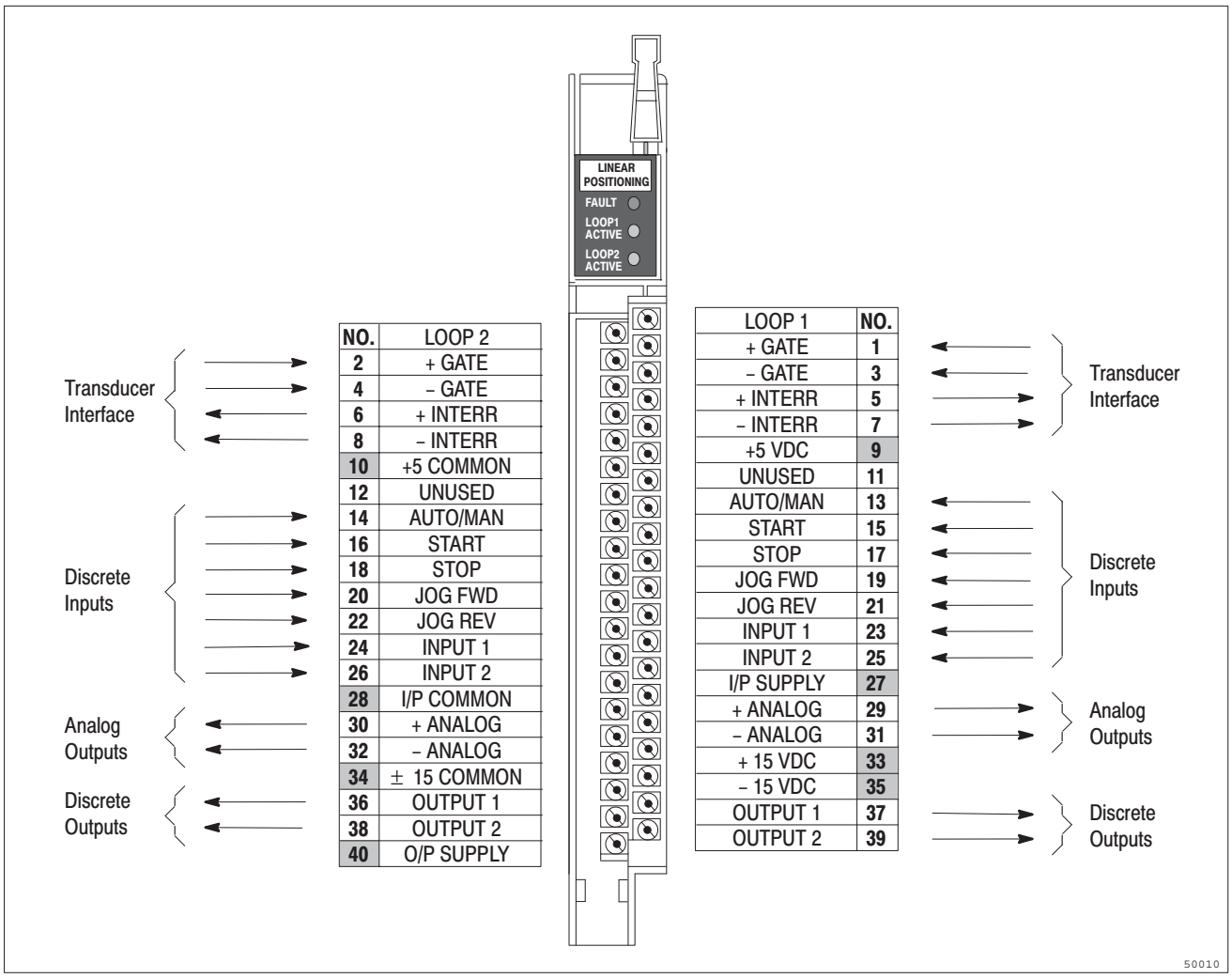
When you first power up the module, all three indicators turn on for about one second. Next, the **LOOP 1 ACTIVE** and **LOOP 2 ACTIVE** indicators turn off while the module performs diagnostics. If the diagnostics discover a module fault, the red **FAULT** indicator stays on and the module remains inactive. When the programmable controller is in run mode, the indicators behave as follows:

- **FAULT** - a red indicator that is normally off. The indicator turns on if there is a module fault in one loop or both loops. See Chapter 11 for more information on module faults.
- L**OOP 1 ACTIVE** - a green indicator that is on when loop 1 is active. The indicator blinks if a fault occurs on loop 1 and turns off if loop 1 is inactive.
- **LOOP 2 ACTIVE** - a green indicator that is on when loop 2 is active. The indicator blinks if a fault occurs on loop 2 and turns off if loop 2 is inactive.

Wiring Arm Terminals

The module draws power for its internal circuitry and communicates with the programmable controller through the 1771 universal I/O chassis. You make all other connections through the wiring arm terminals. Cable length can be up to 200 feet for these connections, depending on the gauge used. See Chapter 5 for wiring guidelines. Figure 4.2 shows the wiring arm terminals for both control loops.

Figure 4.2
Wiring Arm Terminals



The input and output terminals of each of the module's control loops are in four groups. Each group is electrically isolated from the 1771 backplane and from the three other groups:

- transducer interface terminals
- discrete input terminals

- analog output interface terminals
- discrete output terminals

The terminals for these four groups are divided between loop 1 and loop 2. Odd number terminals are for loop 1; even numbered terminals apply to loop 2.

Transducer Interface

Terminals 1 through 8 on the module's wiring arm provide connection points for the transducer interface. The module is designed to work with the linear displacement transducers (LDT) listed in Chapter 1.

The transducer interface circuit is electrically isolated from the 1771 I/O chassis. This protects the 1771 backplane from noise and current surges in the transducer circuits. The transient isolation exceeds 1,500 volts RMS. The transducer interface is also isolated from the other module interfaces and external power supplies.

The module supports a transducer length of up to 15 feet (4572 mm), and can resolve the signal from the transducer to within two thousandths of an inch with one circulation. You can achieve a higher accuracy by configuring the transducer for more circulations. For example, the resolution for 60 inches (1524 mm) is better than one thousandth of an inch if two recirculations are used.

Determining the Optimum Number of Circulations

Every two milliseconds, the module sends an interrogate signal to the transducer. The transducer returns a pulse width that is proportional to the axis position. The maximum pulse width that can be measured without overflowing the counter is about 1680 microseconds (1.680 milliseconds).

The pulse width returned to the module depends on the transducer stroke length and the number of circulations. Each doubling of the number of circulations doubles the width of the gate pulse and the resolution of the position reading. Doubling the gate pulse length, however, effectively halves the maximum transducer length supported by the module, because the maximum pulse width is still determined by the size of the module's counter. Overflowing the counter causes a feedback fault. Is is recommended that you configure the digital interface box for the highest number of circulations that still allows a long enough stroke length for your application. Increasing the number of circulations reduces the effect of noise and improves resolution.

Use these equations to determine the maximum length and positioning resolution for the transducer:

maximum length = 1680/(T x N)
resolution = 1/(58.5 x T x N)

where:

- T = transducer constant stamped on transducer head (typically 9.0500 microseconds per inch)
- N = number of circulations

The following table gives several maximum transducer lengths assuming a transducer constant of 9.0500 microseconds per inch. Resolutions may be limited by the physical capabilities of the transducer. See Chapter 8 for a description of a procedure for verifying the transducer constant.

Number of Circulations	Resolution (Inches)	Maximum Transducer Length (Inches)
1	0.002	185.6
2	0.001	92.8
3	0.0006	61.9
4	0.0005	46.4
5	0.0004	37.1
6	0.0003	30.9
7	0.0003	26.5
8	0.0002	23.2
9	0.0002	20.6
10	0.0002	18.6

Important: Apply a 10% to 20% margin when determining the maximum transducer length. The available stroke length will be less than indicated above due to the null space (typically 2 inches) near the transducer head.

Discrete Inputs

Terminals 13 through 26 on the module's wiring arm provide connection points for discrete input signals. Seven terminals (for each loop) connect to seven discrete inputs.

The use of these inputs is optional. If you do not want to use them, you can disable them through the parameter block. (See Chapter 7.) If you disable the inputs:

- the hardware stop input is deactivated (you do not have to tie it high)
- the auto/manual input defaults to auto
- the programmable controller programs can still read the status of the discrete inputs in the status block

Because the programmable controller programs can still read the status of the discrete inputs, by disabling them you can redefine them for your own purposes.

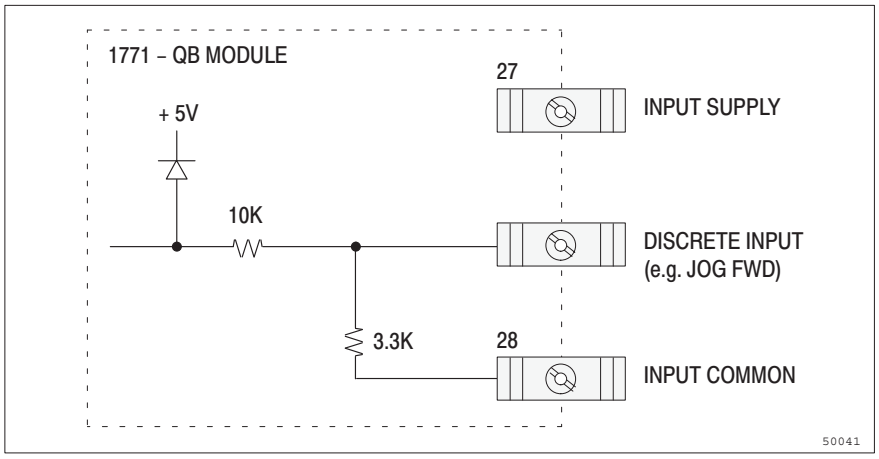
Here are the requirements of the discrete inputs:

low signal	0 to 4 VDC
high signal	10.0 to 30.0 VDC
peak input current	8 mA at 12 VDC 16 mA at 24 VDC

The discrete inputs are configured as current sinks. To reduce heat dissipation, the module turns the discrete input currents off between samples at a 20% duty cycle every 2 ms.

Each discrete input has an internal pull-down resistor. If the device that you have connected to an input provides a high signal, the device must source current through the pull-down resistor. Figure 4.3 is a simplified schematic of a discrete input circuit.

Figure 4.3
Simplified Schematic of a Discrete Input



Auto/Manual Input

The module accepts the signal at the **AUTO/MAN** terminal (13/14) as the auto/manual input. Use this input in conjunction with block transfers to set the operation mode for the axis. A high input means auto mode and a low input means manual mode. The auto/manual input defaults to auto mode if the inputs are disabled via the parameter block.

Important: To set the mode of the axis to auto, you must set both the auto/manual input and the auto/manual bit in the command block high. If either the bit or the input is low, the mode is manual.

Hardware Start Input

In the auto mode, the module accepts a transition from low to high at the **START** terminal (15/16) as a high-true hardware start input signal.

If the axis is in auto mode, and if the hardware start has been enabled via the command block, the module waits for a transition from low to high at the START terminal before it will start axis movement to a previously commanded setpoint. If you don't want to use this feature, disable the hardware start via the command block.

Important: Because of the module's built-in switch debouncing, the low-to-high transition must follow a minimum 16 ms low signal.

Hardware Stop Input

The module accepts the signal at the **STOP** terminal (17/18) as a low-true hardware stop input. A low signal at the hardware stop input disables the analog output and stops axis movement. Unless the discrete inputs are disabled via the parameter block, this input must be high for normal operation. If the connection breaks, axis movement stops.

Example: If the loop fault output of one axis is connected to the hardware stop input of another axis, the movement of both axes will stop if a fault occurs.

Jog Forward Input

In manual mode, the module accepts a high signal at the **JOG FWD** terminal (19/20) as a high-true jog forward signal. When the module receives this signal, it moves the tool or workpiece forward until it reaches the software limit or until the input goes low. Forward is the direction of positive movement relative to the zero-position offset. Chapter 7 explains how to define the zero-position offset in the parameter block.

Jog Reverse Input

In manual mode, the module accepts a high signal at the **JOG REV** terminal (21/22) as a high-true jog reverse signal. When the module receives this signal, it moves the tool or workpiece in the reverse direction until it reaches the software limit or until the input goes low. Reverse is the direction of negative movement relative to the zero-position offset.

Important: If the module detects a feedback fault, the jog inputs will perform open-loop jogs only. This means that the module can send velocity commands to the servo valve (at the low jog rate), but can't monitor axis position. Therefore, software travel limits are ignored.

General Purpose Inputs

There are two general purpose inputs for each control loop of the module at terminals **INPUT 1** (23/24) and **INPUT 2** (25/26). You can monitor the state of the signal at these terminals through the status block. These inputs can also be configured as programmable as described in Chapter 9.

Analog Output Interface

The module's analog outputs, terminals 29 through 32, connect to a hydraulic valve for each axis that the module controls. These outputs supply up to $\pm$100 mA for direct servo valve control or up to $\pm$10 V for proportional valve amplifiers or other voltage controlled devices.

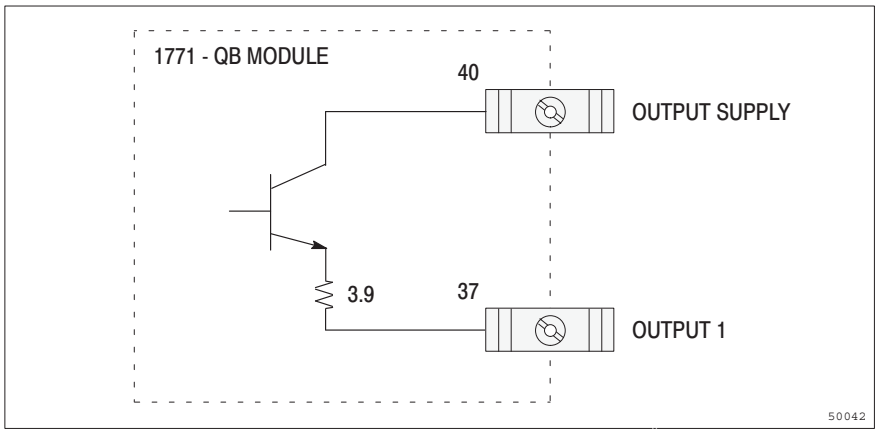
The analog output interface circuit is electrically isolated from the 1771 I/O chassis. This feature protects other devices on the 1771 backplane from noise and current surges in the analog output circuit. An internal relay automatically shuts off these outputs in the event of a module fault. For details on connecting the servo valve interface, see Chapter 5.

Important: Throughout this manual we refer to servo valves, but you can also use the analog outputs to control proportional valves. All references to servo valves also apply to proportional valves.

Discrete Outputs

Terminals 36 through 39 on the module's wiring arm provide connection points for discrete output signals. Each axis has two discrete outputs: Output 1 which can be configured to be either an in-position or programmable output, and Output 2 which can be configured to be either a loop fault or programmable output. (See Chapter 9.) The default configuration is in-position and loop fault. The discrete outputs are current sources. Figure 4.4 gives a simplified schematic of a discrete output circuit.

Figure 4.4
Simplified Schematic of a Discrete Output



Here are the characteristics of the discrete outputs:

Low	no voltage applied to the output
High	output supply voltage applied to output
Maximum Current	100 mA
Voltage Drop	1.6 VDC maximum (at 100 mA) between the discrete output power supply (terminal 40) and the discrete outputs

Important: If you want to connect a discrete output of one axis to the discrete input of another axis, the minimum discrete output supply voltage is 11.6 VDC. This accounts for the voltage drop of 1.6 VDC shown above and provides the minimum voltage required to drive a module discrete input (10 VDC).

ATTENTION: The discrete outputs can withstand a short circuit for a few seconds. However, a continuous short circuit will damage the module's discrete output transistor.

OUTPUT 1

When OUTPUT 1 (terminals 36/37) is configured as an in-position output, it turns off when axis movement toward a commanded endpoint begins and turns on when the axis enters the in-position band (defined in the parameter block). You can connect an in-position output to a hardware start input to provide a simple form of axis coordination.

When this output is configured as a programmable output, its state is determined by the configuration information provided in the motion blocks. (See Chapter 9.)

OUTPUT 2

When OUTPUT 2 (terminal 38/39) is configured as a loop fault output, it is high under normal axis operation. When the module detects a fault in the axis, the loop fault output goes low.

You can connect the loop fault output to the hardware stop input of other control loops so all axis movement will stop if a fault occurs. The loop fault output then provides the low signal required by the hardware stop input of the other axis.

As with OUTPUT 1, OUTPUT 2 can be configured as a programmable output and its state determined by information in the motion blocks.

Power Supplies

You must provide external DC power for the input and output circuits. You could use a single supply, but you'll maintain maximum separation and keep noise to a minimum by using four separate power supplies. In less critical applications, you could power two or three circuits from the same supply.

to power the:	supply:	to these terminals:
Transducer interface	+5 VDC	9, 10
Discrete inputs	+24 VDC (max)	27, 28
Servo valve interface	±15 VDC	33, 34, 35
Discrete outputs	+30 VDC (max)	40

All power connections must be made for the transducer, servo valve, and discrete outputs. The power supply for discrete inputs may be left unconnected if the discrete input disable bit has been set in the parameter block.

Chapter 5

Installing the Linear Positioning Module

Before You Begin

This chapter tells you how to install the module in the I/O chassis and how to configure the module's analog outputs by setting DIP switches. Before you install the module:

- make sure your power supply is adequate
- plan your module's location in the I/O chassis
- take steps to avoid electrostatic discharge

Avoiding Backplane Power Supply Overload

Make sure your power supply can handle the extra load before installing the module in your I/O chassis. Add the module's current requirement, listed on the module's label, to the currents required by other modules inserted in the I/O chassis. If the backplane power supply rating is less than the total current required, you'll need a larger power supply.

Here are the current ratings for the various Allen-Bradley power supply modules.

This Power Supply Module:	Is Rated at:
1771-P1	6.5A
1771-P2	6.5A
1771-P4	8A
1771-P5	8A
1771-P7	16A

Planning Module Location

The module requires one I/O chassis slot. You can install it in any slot in the I/O chassis. The module uses both the output image table byte and the input image table byte that corresponds to its location address.

Electrostatic Discharge

Under some conditions, electrostatic discharge can degrade performance or damage the module. Observe the following precautions to guard against electrostatic damage:

- use a static-free workstation if one is available
- touch a grounded object to discharge yourself before handling the module
- don't touch the backplane connector or connector pins
- when you set the analog output switches, don't touch other circuit components inside the module
- keep the module in a static-shielded bag when it's not in use

Setting Analog Output Switches

You set the analog output DIP switches to define the range of output voltage or output current for the analog output of each control loop.

There are two switch assemblies for each control loop: a single switch assembly that selects voltage or current output and a dual switch assembly that sets the current range. The current range switch has no effect if you choose a voltage output. You must limit the voltage range through the analog range word in the parameter block if you require a voltage range of less than ±10 VDC. (See Chapter 7.)

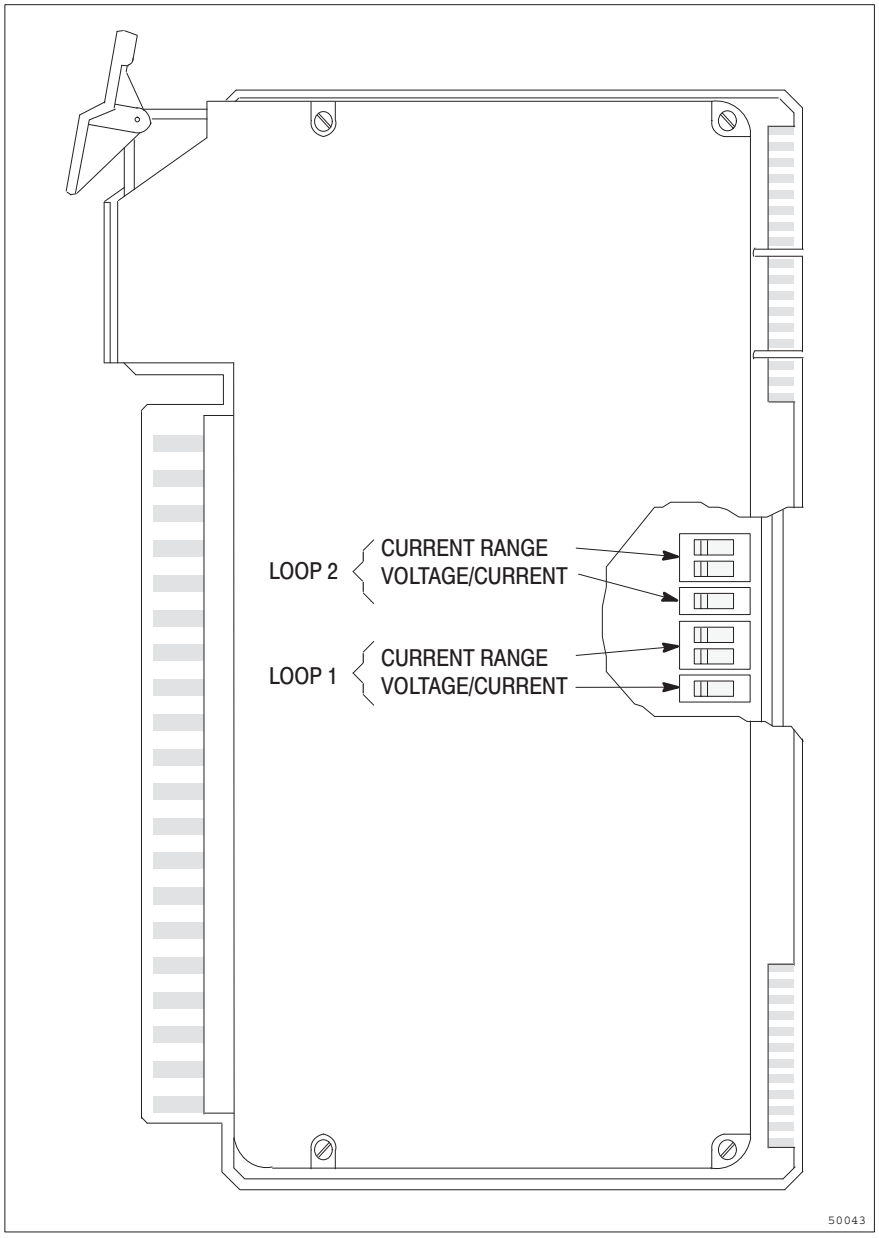
If the analog output will be controlling a current controlled device, such as a servo valve, set the single switch for current and set the current range to match the device. If your device requires a range that falls between those provided, select the next higher range and reduce the range using the analog range word in the parameter block.

Important: Although you can set the current range with the analog range word in the parameter block, you'll improve analog output resolution by first limiting the range with the current range DIP switch.

To set the switches:

1. Lay the module on its side and locate the switches using Figure 5.1. All switches are accessible from the right edge of the module without removing the module cover.

Figure 5.1
Locating the Analog Configuration Switches

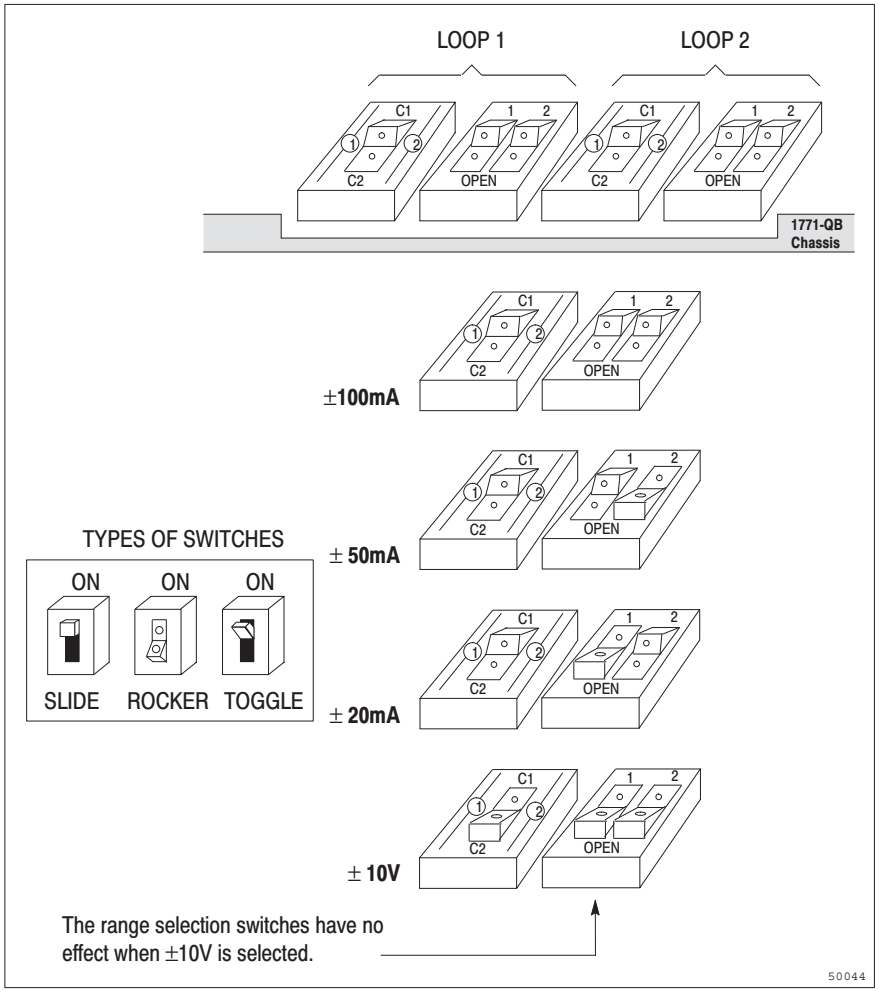


2. Use a blunt pointed instrument (such as a ballpoint pen) to set the switches.

ATTENTION: Don't use a pencil to set switches. Lead can jam the switch.

3. Set the current/voltage switch for each control loop as shown in Figure 5.2.

Figure 5.2
Configuring the Analog Outputs



4. If you have selected a current output, set the current range switch to ±100 mA, ±50 mA, or ±20 mA as shown in Figure 5.2. If your device requires a range that falls between those provided, select the next higher range and reduce the range using the analog range word in the parameter block.

ATTENTION: If your switch setting does not provide enough current, the servo valve may not operate to its full capability. On the other hand, excessive currents may damage the servo valve.

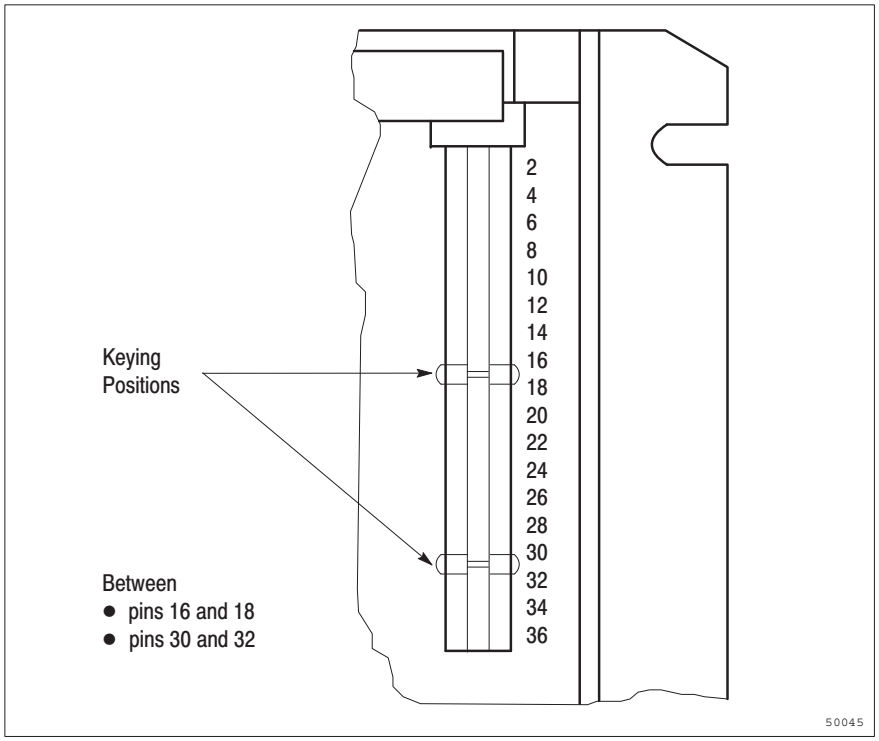
Keying

A package of plastic keys (Cat. No. 1771-RK) is provided with every I/O chassis. When properly installed, these keys prevent the seating of anything but the module in the keyed I/O chassis slot. Keys also help to align the module with the backplane connector.

Each module is slotted at its rear edge. Position the keys on the chassis backplane connector, corresponding to the slots on the module's rear edge.

Insert the keys into the upper backplane connectors. Position the keys between the numbers at the right of the connectors, as shown in Figure 5.3.

Figure 5.3
Module Keying



Inserting the Module

After setting analog output switches and setting the keying positions, you're ready to insert the module into a slot in the I/O chassis.

To insert the module, follow this procedure:

1. Remove all power from the I/O chassis and from the module's wiring arm before inserting or removing a module.

2. Open the module locking latch on the I/O chassis and insert the module into the slot keyed for it.

3. Press firmly to seat the module into the backplane connector.

4. Secure the module with the module locking latch.

ATTENTION: Don't force a module into the backplane connector. If you can't seat a module with firm pressure, check the alignment and keying. Forcing a module can damage the backplane connector and the module.

Wiring Guidelines

Through the module's terminals, you connect the module to external devices. The exact wire gauge and maximum allowable length depends upon the devices being connected. Here are some general rules to follow when you connect the terminals:

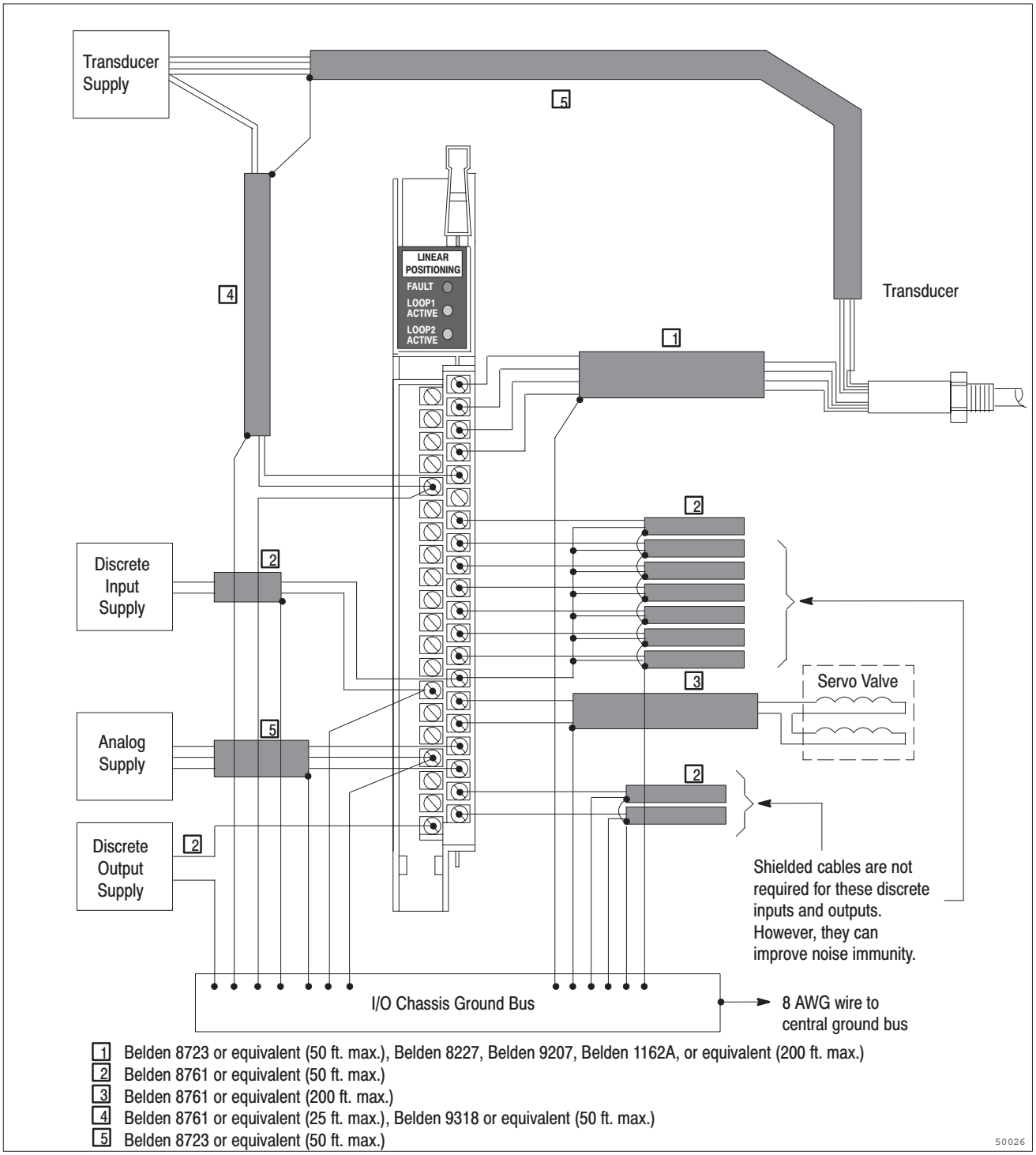
- don't use wire with too large a gauge. The maximum practical wire gauge is 14 AWG.
- keep low level conductors separate from high level conductors. Follow the practices outlined in Publication 1770-980 P2LC *Grounding and Wiring Guidelines*.
- keep your power supply cables as short as possible–less than 50 feet is preferable.

Using Shielded Cables

For many connections, you are instructed to use shielded cables. Using shielded cables and properly connecting their shields to ground protects against electromagnetic noise interfering with the signals transmitted through the cables. Connect each shield to ground at *one and only one end*. At the other end, cut the shield foil and drain wire short and cover them with tape. This will protect them against accidentally touching ground. Keep the length of leads extending beyond the shield as short as possible.

Figure 5.4 shows shielded cable connections for one control loop. Mount a ground bus directly below the I/O chassis to provide a connection point for the cable shield drain wires and the common connections for the input and output circuits. Connect the I/O chassis ground bus through 8 AWG wire to the central ground bus to provide a continuous path to ground.

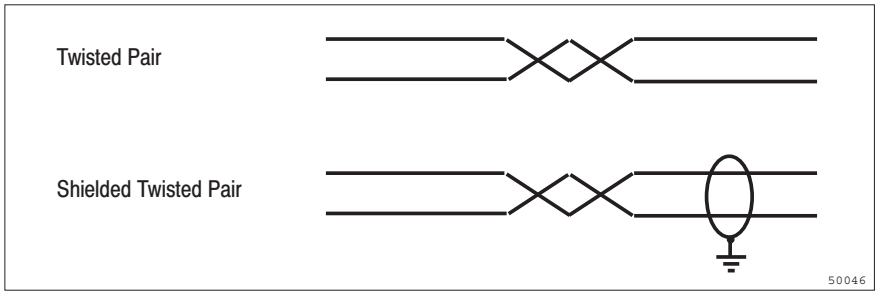
Figure 5.4
Shielded Cable Grounding Connections


Transducer Supply
5
LINEAR POSITIONING
FAULT
LOOP1 ACTIVE
LOOP2 ACTIVE
4
1
Transducer
2
Discrete Input Supply
2
3
Servo Valve
Analog Supply
5
2
Discrete Output Supply
2
Shielded cables are not required for these discrete inputs and outputs. However, they can improve noise immunity.
I/O Chassis Ground Bus
8 AWG wire to central ground bus
1 Belden 8723 or equivalent (50 ft. max.), Belden 8227, Belden 9207, Belden 1162A, or equivalent (200 ft. max.)
2 Belden 8761 or equivalent (50 ft. max.)
3 Belden 8761 or equivalent (200 ft. max.)
4 Belden 8761 or equivalent (25 ft. max.), Belden 9318 or equivalent (50 ft. max.)
5 Belden 8723 or equivalent (50 ft. max.)
50026

Using Twisted Wire Pairs

It is recommended you use twisted wire pairs for a signal and its return path to reduce noise levels further. Figure 5.5 shows a twisted pair and shielded twisted pair.

Figure 5.5
Shielded Twisted Pair Diagram



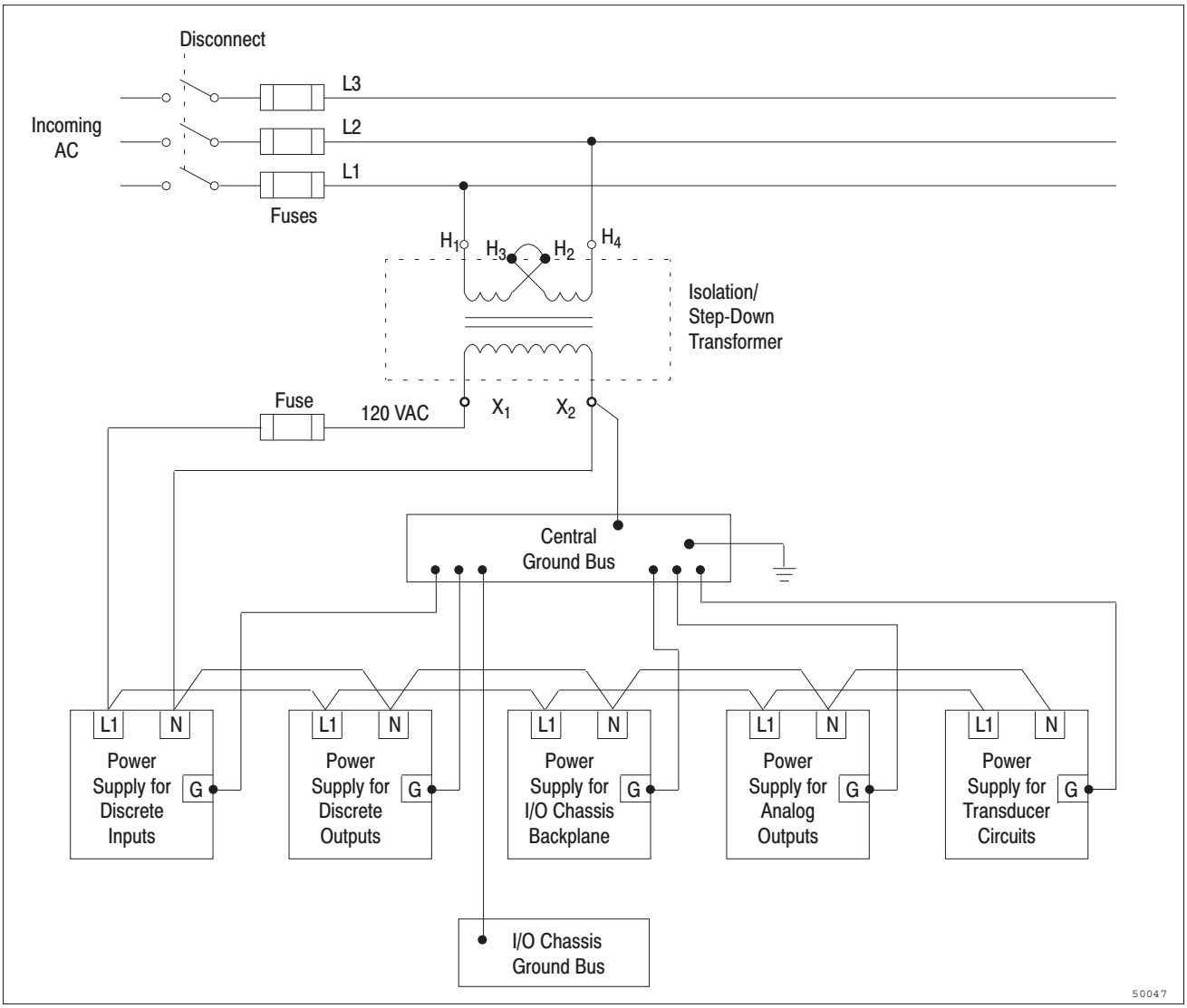
ATTENTION: Failure to follow correct shielding procedures can cause unpredictable movement resulting in possible injury to personnel and damage to equipment.

Connecting AC Power

Figure 5.6 shows AC power and ground connections. Incoming AC connects to the primary of an isolation transformer. The secondary of the isolation transformer connects to:

- the power supply for the discrete inputs
- the power supply for the discrete outputs
- the power supply for the I/O chassis
- the power supply for the analog outputs
- the power supply for the transducer circuits

Figure 5.6
AC Power and Ground Connections



In the grounded AC system shown above, the low side of the isolation transformer is connected to the central ground bus. Figure 5.6 also shows connections from the central ground bus to each power supply and to the I/O chassis ground bus shown in Figure 5.4.

Power Supplies

The 1771 backplane provides the power for most of the module circuits. You'll need external power supplies for the analog outputs, transducer interfaces, discrete inputs and discrete outputs.

All four power supplies and their associated module circuits are electrically isolated from the I/O chassis and from each other. To provide maximum isolation of the four sets of circuits, the four supplies should be from separate sources. However, you can use the same power supply to power two or more circuits if you don't need the isolation that separate supplies provide.

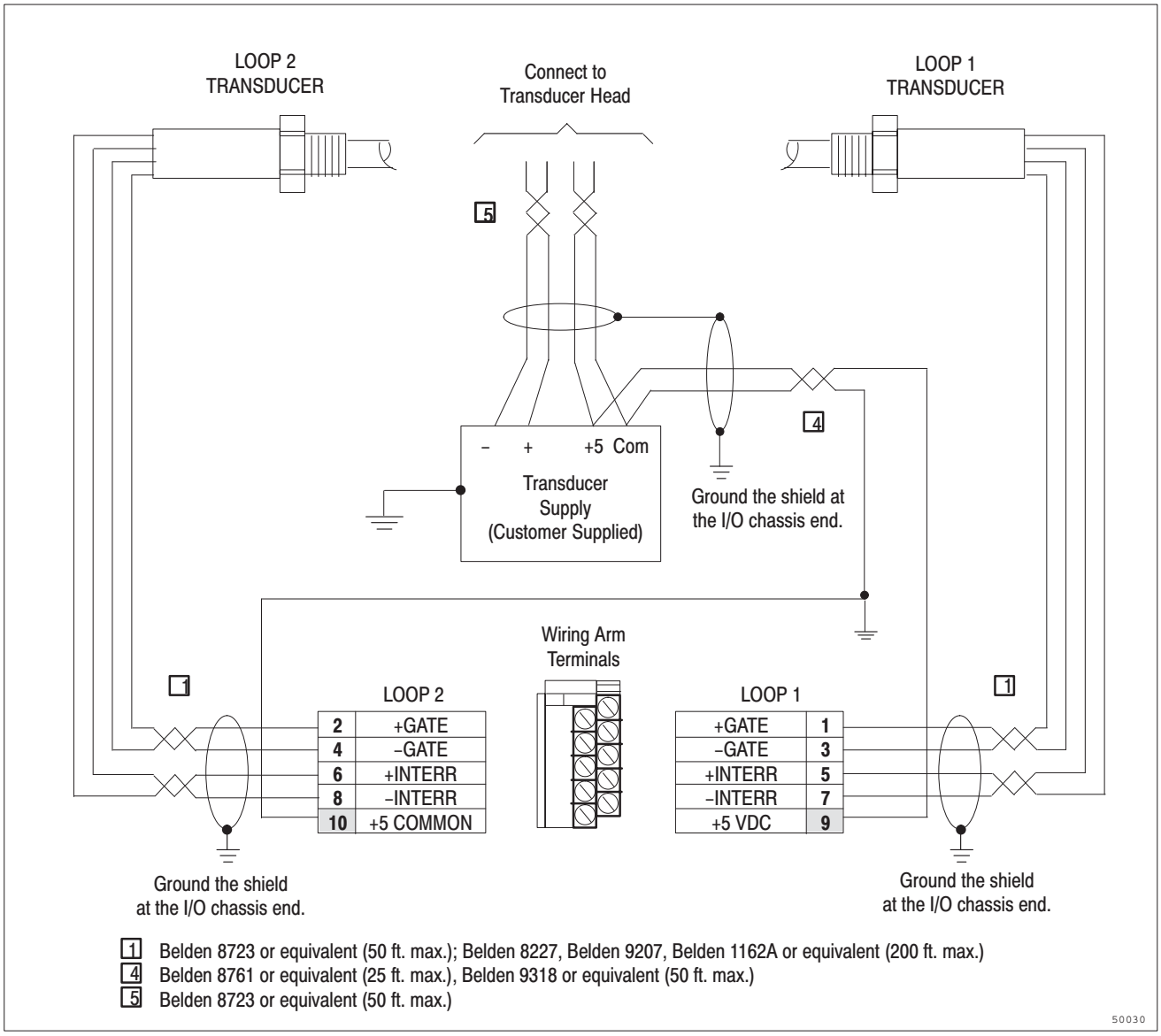
Information on how to connect the power supply for each circuit is under the heading for that circuit.

Connecting the Transducer Interface

Figure 5.7 shows the transducer interface connections. You should refer to the wiring diagrams supplied with your transducer to determine pinouts on the transducer head.

Important: The transducer must be configured for external interrogation.

Figure 5.7
Transducer Connections



Power Supply

To connect the transducer power supply, follow these steps:

1. Connect +5 VDC from your power supply to the +5 VDC terminal (9) on the module.

2. Connect + VDC from your power supply to the transducer.

3. Connect - VDC from your power supply to the transducer.

4. Connect the common terminal on your power supply to the +5 COMMON terminal (10) on the module, to ground at the I/O chassis, and to the transducer.

5. Connect the cable shields to ground at the I/O chassis end.

6. Connect the power supply chassis to ground.

Transducer Interface

After connecting the transducer power supply to the module, make the Gate and Interrogate connections. Use a single, continuous, shielded cable segment for these connections. Don't break the cable for connection in a junction box, but connect it directly from the digital interface box to the module.

To connect the transducer interface terminals:

1. Configure the transducer for external interrogation.

2. If you haven't already configured the transducer for the optimum number of circulations, do so now. Refer to Chapter 4 for a procedure to determine the optimum number of circulations for your system.

3. Connect the module's +GATE terminal (1/2) to the transducer's +GATE terminal.

4. Connect the module's -GATE terminal (3/4) to the transducer's –GATE terminal.

5. Connect the module's +INTERR terminal (5/6) to the transducer's +INTERROGATE terminal.

6. Connect the module's -INTERR terminal (7/8) to the transducer's –INTERROGATE terminal.

7. Connect the cable shields to ground at the I/O chassis end.

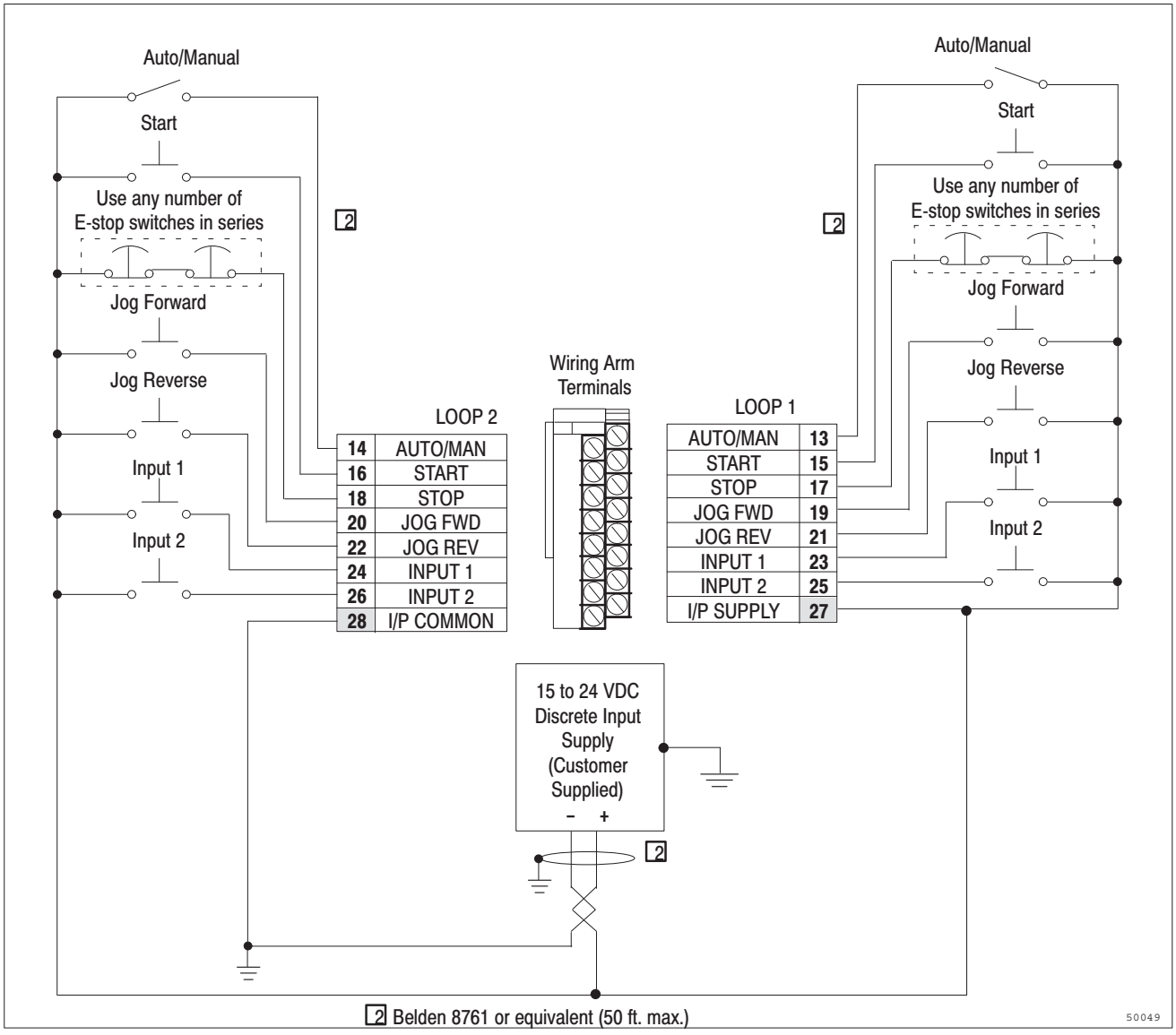
Connecting the Discrete Inputs

The seven discrete inputs (for each control loop) make connections via eight wiring arm terminals (one terminal is discrete input common). The voltage and current requirements for the discrete inputs are:

Low	0 to 4 VDC
High	10 to 30 VDC
Input Current	8 mA @ 12 VDC 16 mA @ 24 VDC

Make sure that the voltage driving each input is at the appropriate level. Figure 5.8 shows the discrete input connections.

Figure 5.8
Discrete Input Connections



If you are driving a discrete input from a discrete output of another module, keep in mind that you must measure the output voltage at the discrete output itself and not at the discrete output power supply. There is a 1.6 VDC drop between the power supply and the discrete output at maximum current. To yield the minimum 10 VDC at the discrete output, the discrete output supply must be greater than 11.6 VDC.

Power Supply

To connect the discrete input power supply, follow these steps:

1. Connect the (+) side of the discrete input power supply to the I/P SUPPLY terminal (27) of the module.
2. Connect the common of the discrete input power supply to the I/P COMMON terminal (28) of the module.
3. Connect the cable shield to ground at the I/O chassis end.
4. Connect the power supply chassis to ground.

Auto/Manual Input

The auto/manual input, in conjunction with the auto/manual bit in the command block, determines the module's mode of operation. Both the auto/manual input and the auto/manual bit must be on to achieve auto mode. Otherwise, the mode is manual. Connect the auto/manual and common terminals to an external source.

Hardware Start Input

The hardware start input performs the same function as the software start. (See Chapter 8.) When you command a setpoint, no axis movement occurs on that control loop until the module receives a start command or a transition from low to high (after at least 20 msec of low) at the control loop START terminal (15/16).

Example: To start movement to a setpoint when Output 1 (configured as the in-position output) of another control loop goes high, connect the hardware start input to the Output 1 of that control loop.

You don't need to connect the hardware start terminals if you won't be using this feature.

Hardware Stop Input

A low input at the STOP terminal (17/18) will stop axis movement for the corresponding control loop. This allows you to connect any number of normally closed emergency stop switches in series between a high source and the hardware stop terminal. Opening any of these switches will immediately zero the analog output for that loop.

ATTENTION: In servo valve control systems, axis drift may occur due to imprecise valve nulling even with zero analog output. It is recommended that emergency stop switches, such as overtravel limit switches, also turn off axis power and close a blocking valve installed between the servo valve and the prime mover.

Important: If you have enabled the discrete inputs via the parameter block, don't leave the hardware stop terminals disconnected–you must connect them to a source which is normally high. If the connection breaks, the input goes low and axis movement automatically stops.

ATTENTION: Use the hardware stop to disable the servo valve drive or stop axis motion only in an emergency. Abruptly stopping axis motion places mechanical stress on the positioning assembly. Use the slide stop bits in the command block to stop axis motion in non-emergencies. The slide stop decelerates before stopping and is less abrupt than the hardware stop.

Jog Forward Input

Before the module responds to the jog forward input, the control loop must be in manual mode.

If you apply high input (more than 10 VDC) to the jog forward input, the axis moves in the forward direction (the direction of positive movement relative to the zero-position offset). It continues to move until the jog forward input is low or until the axis reaches the software travel limit, whichever occurs first.

Chapter 8 explains how to define the zero-position offset in the parameter block.

To set up the jog forward input:

1. Connect a normally open push-button switch between the JOG FWD input terminal (19/20) and the discrete input power supply's positive terminal (27).

2. Mount the jog forward switch so an operator can see the axis motion.

Leave the jog forward input terminal disconnected if you're not using it.

Jog Reverse Input

The jog reverse input is valid only in the manual mode. The jog reverse input is similar to the jog forward input, except the axis movement is in the reverse direction (the direction of negative movement relative to the zero-position offset). Connect the JOG REV terminal (21/22) in the same way as the jog forward input. Leave the terminal disconnected if you are not using it.

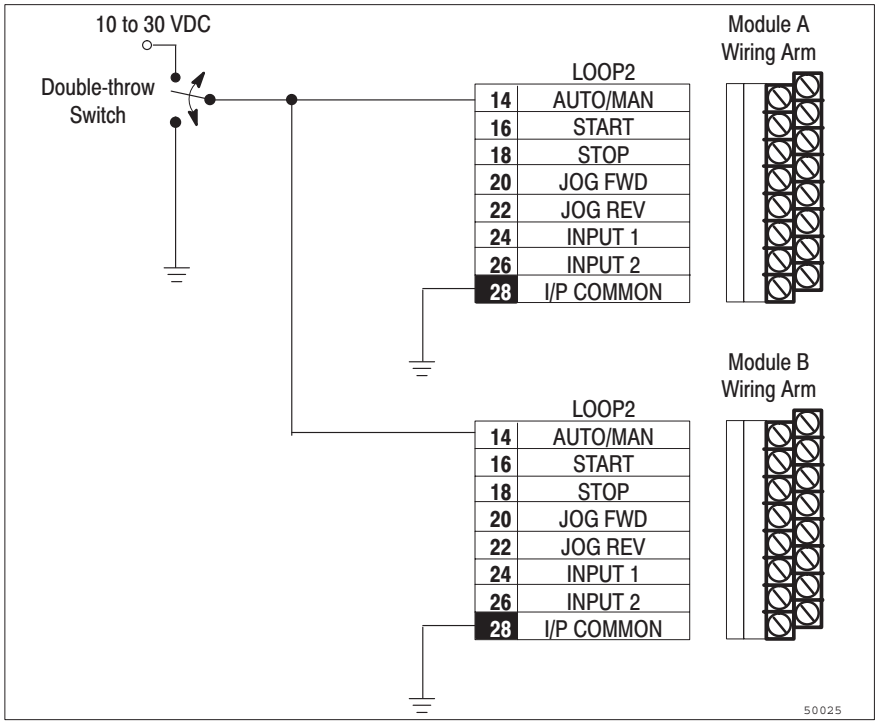
General Purpose Inputs

There are two general purpose inputs for each control loop of the module at terminals INPUT 1 (23/24) and INPUT 2 (25/26). You can monitor the state of the signal at these terminals through the status block (see Chapter 6) or use these terminals as programmable inputs (see Chapter 9).

Connecting Multiple Modules

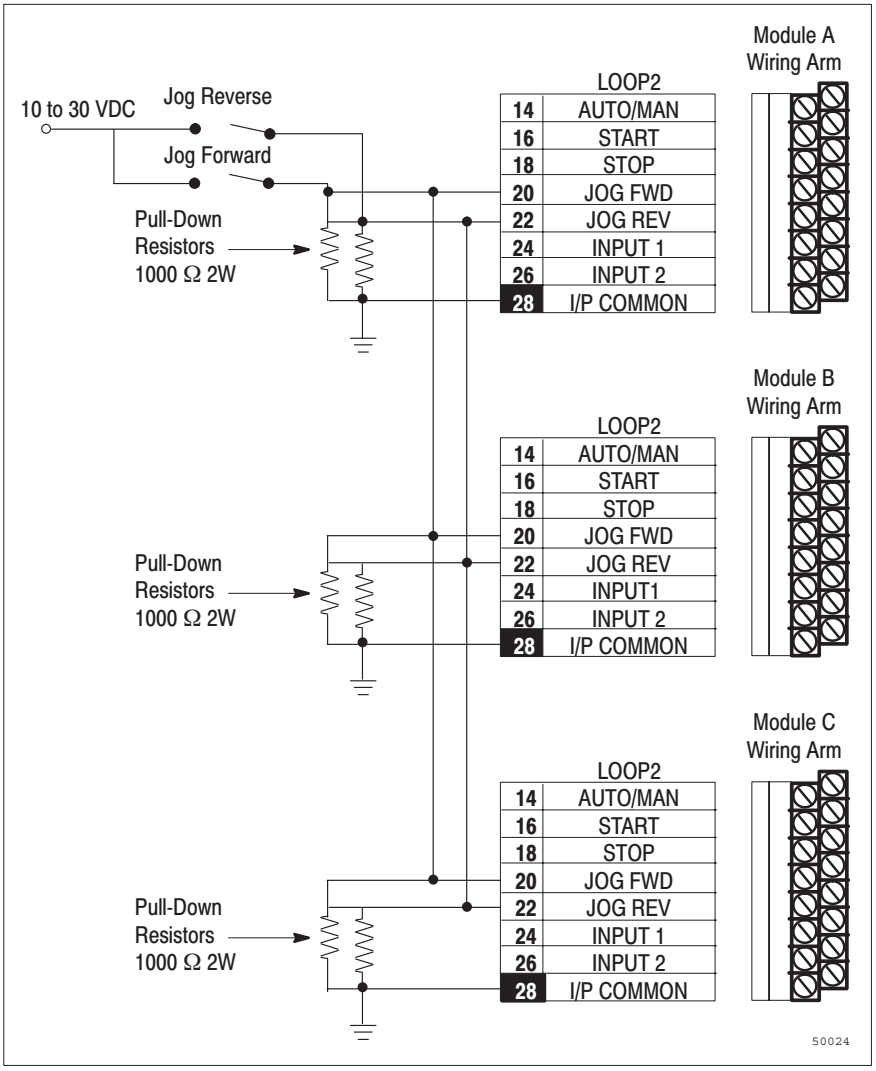
To connect the discrete inputs of two or more modules to a single control line, you must pull the signal to ground with either a double-throw switch or a pull-down resistor.

Figure 5.9
Using a Double-Throw Switch to Control Multiple QB's



Pull-down resistors or double-throw switches are only required if you wish to connect two or more QB's. They are not required to control multiple discrete inputs on a single module.

Figure 5.10
Using Pull-Down Resistors to Control Multiple QB's



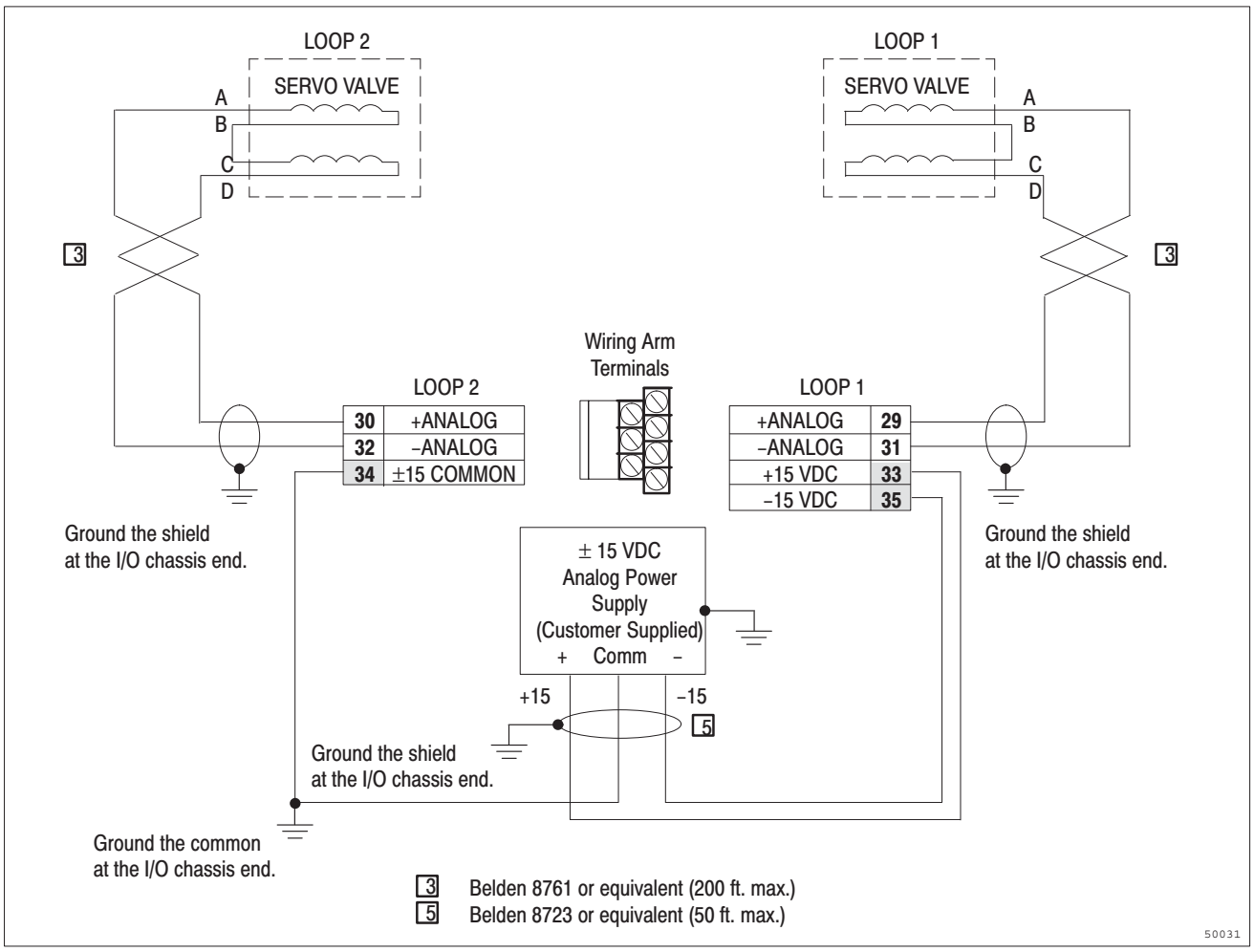
ATTENTION: Failure to follow these procedures can result in sporadic operation of the discrete inputs.

Connecting the Analog Outputs

The analog outputs provide the current (or voltage) by which the module controls the servo valve. By controlling the servo valve, the module controls axis motion.

ATTENTION: Applying output to an axis with polarity reversed can cause sudden high-speed motion. For maximum safety, leave the analog outputs disconnected and the axis power off until you perform the axis tuning procedures in Chapter 8.

Figure 5.11
Analog Output Connections



ATTENTION: The polarity of the analog outputs is affected by the setting of the most significant bit of the analog range words in the parameter block. (See Chapter 7.) Incorrect wiring of the analog outputs or an incorrect setting of this most significant bit can cause the axis to accelerate out of position when the loop is closed.

Power Supply

To connect the analog output supply:

1. Connect the (+) side of the power supply to the +15 VDC module terminal (33).
2. Connect the (-) side to the -15 VDC module terminal (35).
3. Connect the common to ±15 COMMON module terminal (34) and to ground at the I/O chassis.
4. Connect the shield to ground at the I/O chassis end.
5. Connect the analog power supply chassis to ground.

Analog Output

To connect the analog output of the control loop to the servo valve interface:

1. Be sure that you set the control loop's voltage/current selection switches to match your servo valve's requirements.
2. Check that the analog output power supply is connected.
3. Connect the +ANALOG module terminal (29/30) and the -ANALOG module terminal (31/32) to the servo valve coil terminals.

Important: If you select voltage output for a loop, the module internally connects that loop's -ANALOG terminal to the ±15 COMMON terminal (34).

4. Connect the cable shields to ground at the I/O chassis end.

Important: Wire servo valve coils in series. Refer to the instructions for your device.

Connecting the Discrete Outputs

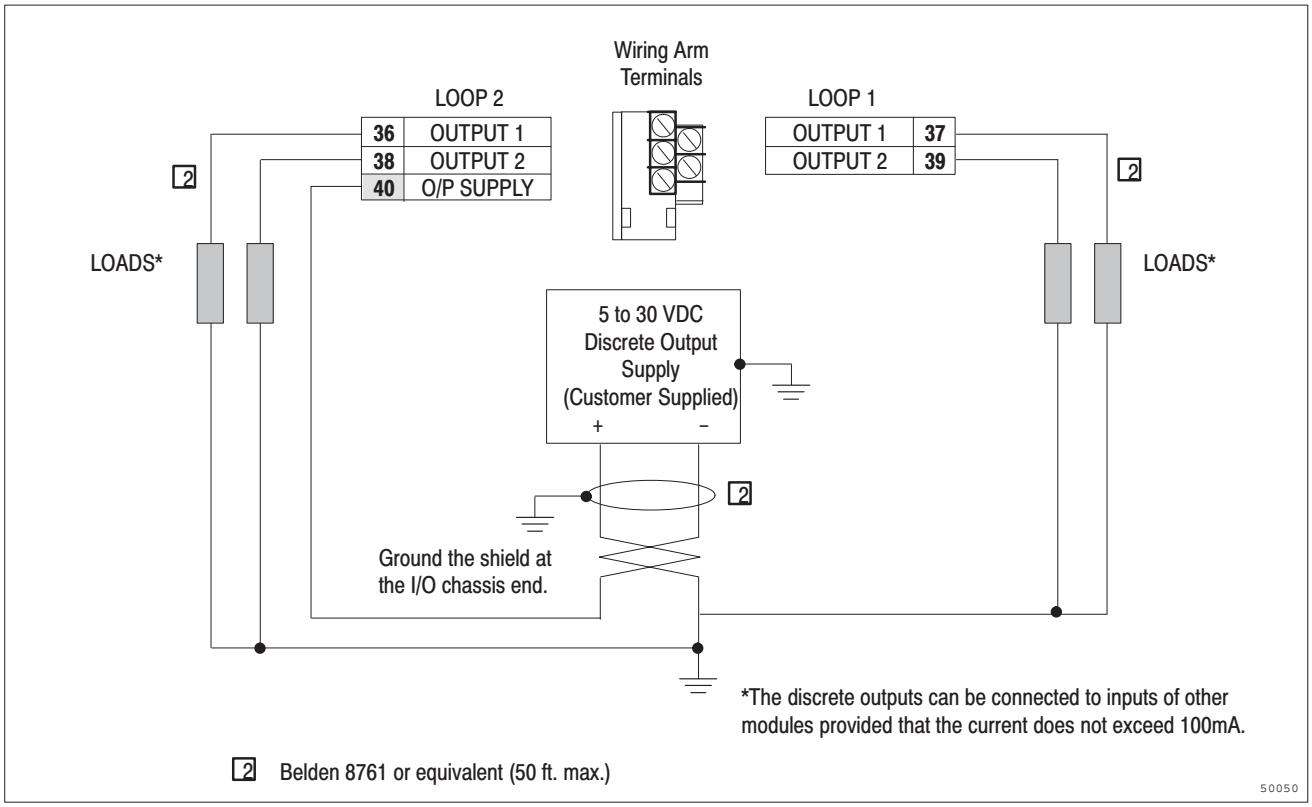
The two discrete outputs for each loop are powered by the discrete output power supply. The characteristics of the discrete outputs are:

Low	no voltage applied to the output
High	output supply voltage applied to output
Maximum Current	100 mA
Voltage Drop	1.6 VDC maximum (at 100 mA) between the discrete output power supply (terminal 40) and the discrete outputs

Power for the OUTOUT 1 and OUTPUT 2 discrete outputs comes from the discrete output power supply through terminal 40 on the module wiring arm. The return for the load driven by OUTOUT 1 or OUTPUT 2 connects to the common of the discrete output power supply.

Figure 5.12 shows a typical discrete output connection.

Figure 5.12
Discrete Output Connections


Wiring Arm Terminals
LOOP 2
36 OUTPUT 1
38 OUTPUT 2
40 O/P SUPPLY
LOOP 1
OUTPUT 1 37
OUTPUT 2 39
2
LOADS*
LOADS*
5 to 30 VDC Discrete Output Supply (Customer Supplied)
+
–
Ground the shield at the I/O chassis end.
*The discrete outputs can be connected to inputs of other modules provided that the current does not exceed 100mA.
2 Belden 8761 or equivalent (50 ft. max.)
50050

Power Supply

To connect the discrete output power supply, follow these steps:

1. Connect the (+) side of the discrete output power supply to the O/P SUPPLY terminal (40) on the module.
2. Connect the common of the discrete output power supply to ground at the I/O chassis and to the returns (-) of all output devices.
3. Connect the discrete output power supply chassis to ground.
4. Connect the shield to ground at the I/O chassis end.

OUTPUT 1

If configured as an in-position output, this output goes low in auto mode when the axis begins to move towards a commanded setpoint or after a jog. It goes high when the axis enters the in-position band surrounding an endpoint.

You can use this output to drive another control loop to coordinate the axis movement of various control loops. In this case, you connect the OUTPUT 1 terminal to the discrete hardware start input of another control loop. Otherwise, you may connect this output to a light emitting diode or other indicator.

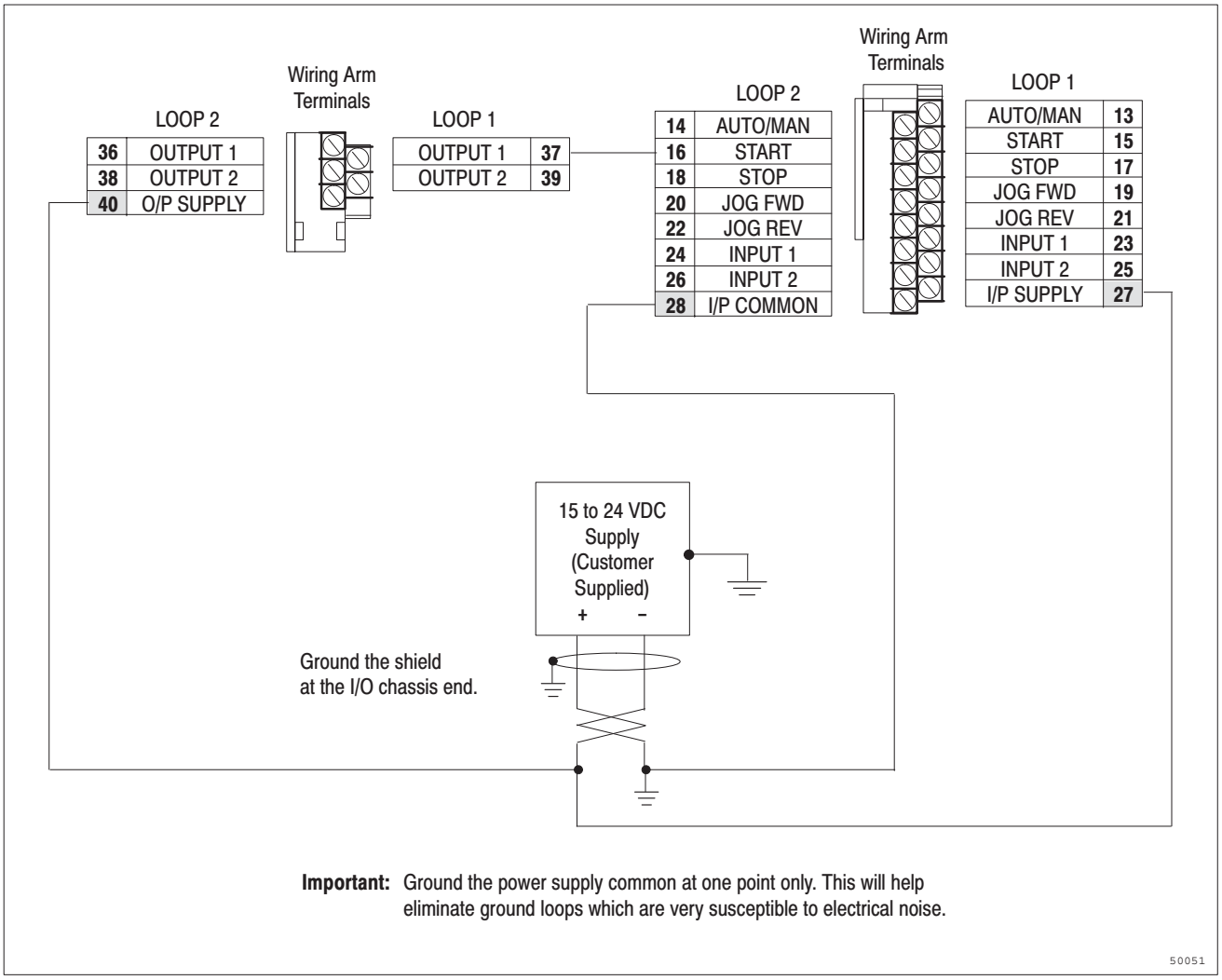
Important: When you are connecting the module's discrete inputs and outputs to external devices, keep in mind that the discrete inputs sink current and the discrete outputs source current.

OUTPUT 2

If configured as a loop fault output, this output is normally high. It goes low when the module detects a fault in the control loop. You can use the loop fault output to drive another module's hardware stop input. You can connect the loop fault output terminal to the hardware stop input of another control loop or to a visual/audible fault indicator.

Both OUTPUT 1 and OUTPUT 2 can be configured to be programmable outputs. (See Chapter 9.)

Figure 5.13
Connecting a Discrete Output to a Discrete Input



Important: Ground the power supply common at one point only. This will help eliminate ground loops which are very susceptible to electrical noise.

Chapter 6

Interpreting Module-to-PLC Data (READS)

This chapter explains how to monitor module operation from a programmable controller by reading and interpreting status block data that the module transfers to the programmable controller's data tables.

PLC Communication Overview

You must program the programmable controller to communicate with the Linear Positioning Module through block read and block write instructions. The data blocks are:

- status block
- parameter block
- setpoint block
- motion block
- command block

The block read instruction transfers the status block data from the module to the programmable controller data table. The block write instruction transfers the parameter block, the setpoint block, the motion block and the command block data from the programmable controller data table to the module. This chapter tells you how to interpret the status block data. Chapter 7 tells you how to format the parameter block, the setpoint block, and the command block data. Chapter 9 explains the motion block.

Status Block

The status block contains information on the status of each axis. Until the module receives a parameter block, the status block consists of five words (i.e. the default assumption of one axis). The size of subsequent status blocks depends on the configuration you program through the parameter block.

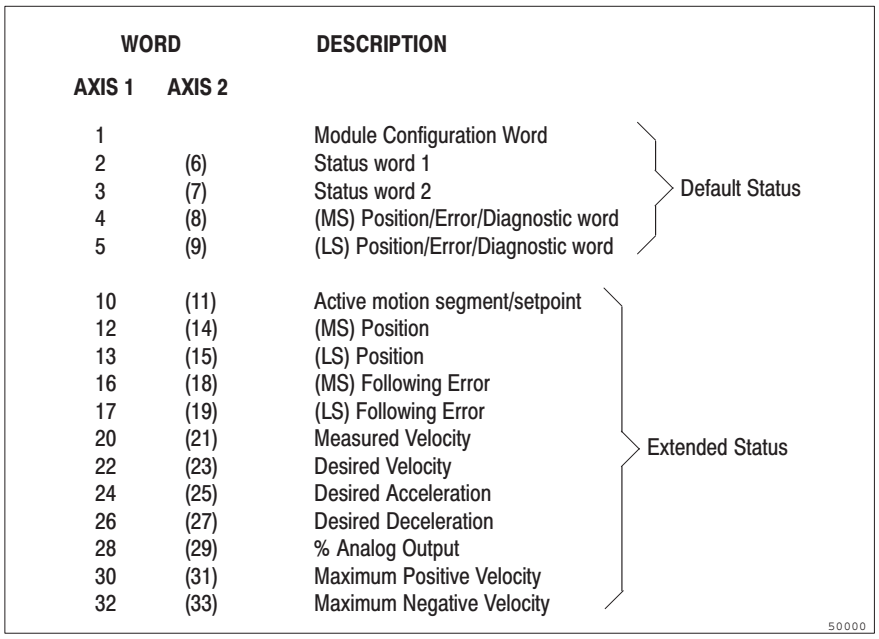
Number of Axes	Status Block Length
1	5 words – default
2	9 words – default
1 or 2	up to 33 words depending on block transfer length requested

You can set the block transfer read instruction to include extended status information by specifying a block transfer length of 33. If you specify a length of 0 (or 64), the module returns the default: 5 words for one axis and 9 for two.

Word Assignment

The assignment of the words within the status block is as follows:

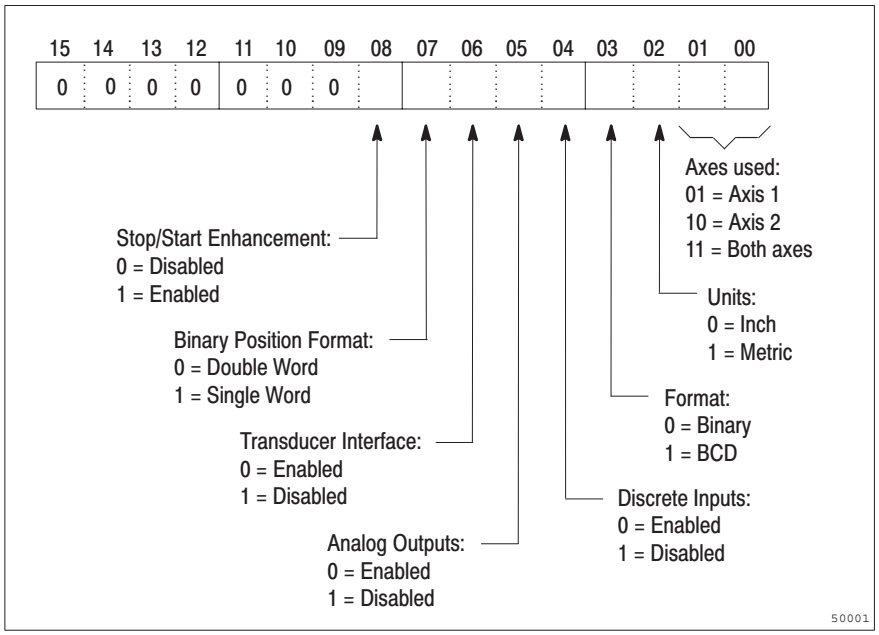
Figure 6.1
Status Block Word Assignments



Module Configuration Word (word 1)

Bits 0 to 8 are controlled by the parameter control word in the parameter block. Module configuration information includes number of axes, units of measurement, number format, binary position format (single or double word), and the state (enabled or disabled) of the start/stop enhancement, discrete input, analog output and transducer interface bits. Detailed descriptions of these are in Chapter 7.

Figure 6.2
Module Configuration Word



Status Word 1 (words 2 and 6)

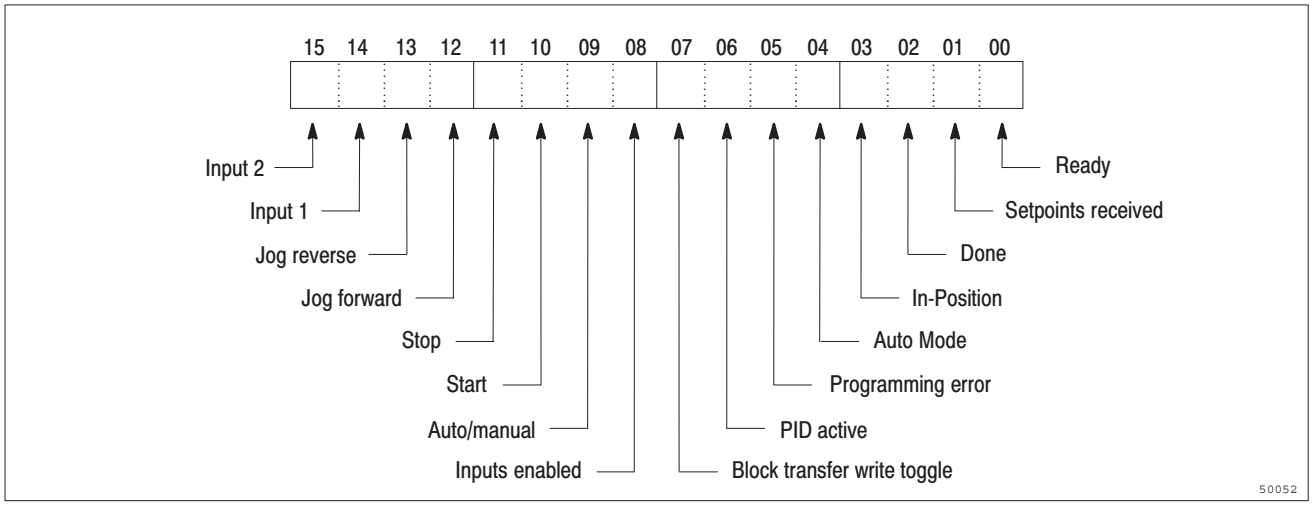
Each bit in status word 1 corresponds to a particular axis condition.

Bit 0 – Ready

The module turns off the ready bit after powerup or after a reset command. (See Chapter 7.) The module turns this bit on when it receives a valid parameter block for this loop. Unless it detects a fault, the module enables the analog output for the axis when the ready bit turns on.

The module does not accept setpoint, motion or command blocks until the ready bit is on.

Figure 6.3
Status Word 1



Bit 1 – Setpoints Received

The setpoints received bit is off after powerup or after a reset command. It turns on after the module receives a valid setpoint block for the loop.

Bit 2 – Done

The module turns the done bit on when the module has finished traversing the axis velocity profile. At this point, the desired velocity is zero and the position command is stabilized at the target endpoint. The module turns this bit off when you command a new setpoint move or jog.

Important: If this bit turns on, it does not mean that the axis is in position yet.

Bit 3 – In-Position

The in-position bit turns on when the done bit is on and the following error has closed to within the in-position band defined in the parameter block. When the in-position bit is on, the axis has moved to within a specified distance of the programmed endpoint.

If configured as the in-position output, then the OUTPUT 1 hardware output reflects the status of this bit. The in-position bit turns off when the axis receives a jog command or begins a move to a new setpoint.

Important: When OUTPUT 1 turns off, it remains off for at least 16 milliseconds. This provides compatibility with the hardware start input.

Bit 4 – Auto Mode

The auto mode bit turns on when the loop is in auto mode, i.e., when the auto/manual bit in the command block is on and the auto/manual hardware input is true. The auto/manual hardware input is true if the module is receiving a high input at the AUTO/MAN discrete input terminal (13/14) or if the discrete inputs are disabled in the parameter block.

Don't confuse the auto mode bit with the auto/manual bit (bit 9). The auto/manual bit simply reflects the state of the signal at the AUTO/MAN discrete input terminal (13/14), regardless of whether or not the discrete inputs are disabled.

Bit 5 – Programming Error

If the module detects an illegal bit combination, such as a non-BCD value where it expects a BCD value, it turns on the programming error bit. You can get additional information from words 4, 5 and 8, 9 if they are configured to display diagnostics. The programming error bit clears when the error condition ends.

Bit 6 – PID Active

The PID bit is on when the integral and derivative terms are enabled. It turns off during axis movement in response to a setpoint or jog command.

Bit 7 – Block Transfer Write Toggle

When the module receives and successfully decodes a block transfer whose block contents or size differ from the previous valid block transfer received, the block transfer write bit is toggled. As long as no programming error occurs, any valid block transfer received by the module will toggle this bit. This lets you synchronize block transfers and ensure that every block transfer sent to the module has been received.

Bit 8 – Inputs Enabled

The inputs enabled bit is off after powerup or after a reset command. It turns on if you enable the discrete inputs by the parameter block. If the discrete inputs are disabled, their status is still displayed in bits 9 through 15 of this status word, but their functions are disabled.

Bit 9 – Auto/Manual

The auto/manual bit reflects the state of the auto/manual hardware input (0 = manual mode, 1 = auto mode).

Bit 10 – Start

The start bit reflects the state of the hardware start input (0 = no start, 1 = start).

Bit 11 – Stop

The stop bit reflects the state of the hardware stop input (0 = stop, 1 = no stop).

Important: The hardware stop is a low-true signal.

Bit 12 – Jog Forward

The jog forward bit reflects the state of the jog forward hardware input (0 = no jog, 1 = jog).

Bit 13 – Jog Reverse

The jog reverse bit reflects the state of the jog reverse hardware input (0 = no jog, 1 = jog).

Bit 14 – Input 1

This bit reflects the state of hardware input 1 (0 = off, 1 = on).

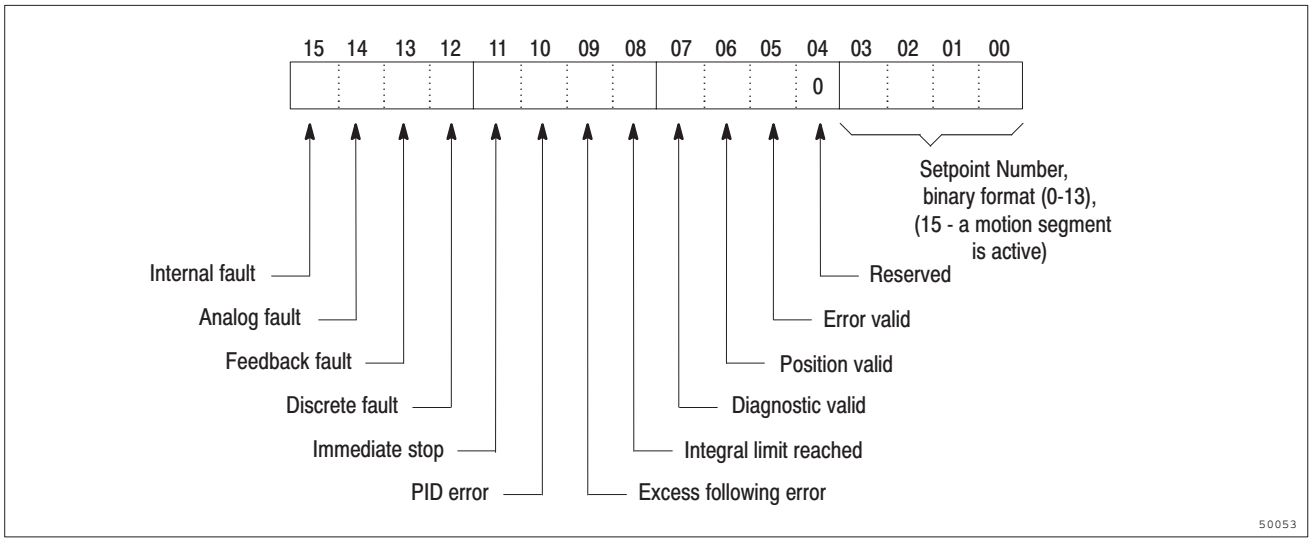
Bit 15 – Input 2

This bit reflects the state of hardware input 2 (0 = off, 1 = on).

Status Word 2 (words 3 and 7)

Status word 2 gives the active setpoint and provides additional status information.

Figure 6.4
Status Word 2



Bits 0 to 3 – Setpoint Number

Bits 0 to 3 give the currently active setpoint (1 to 13 in binary format), or indicate that a motion segment is active (15 binary). This parameter defaults to zero on powerup or after a jog command. It then remains zero until a setpoint or motion start is received and accepted.

The currently active setpoint is the target point of the latest initiated axis move. If hardware started is enabled, the module won't update the setpoint number to a commanded setpoint until after it receives a hardware start.

Bit 4 – Reserved

Bit 4 is reserved for future use.

Bit 5 – Error Valid

The error valid bit is on if the next two status block words (i.e., words 4 and 5 for axis 1 and words 8 and 9 for axis 2) for this axis contain a valid following error value.

Bit 6 – Position Valid

The position valid bit is on if the next two status block words (i.e., words 4 and 5 for axis 1 and words 8 and 9 for axis 2) contain a valid axis position.

Bit 7 – Diagnostic Valid

This bit is on if the next two status block words (i.e., words 4 and 5 for axis 1 and words 8 and 9 for axis 2) for this axis contain diagnostic information.

Bit 8 – Integral Limit Reached

The integral limit reached bit turns on if the integral term of the PID algorithm reaches the maximum specified in the parameter block. (See Chapter 7.) It turns off when the integral term returns to within the permitted limits. Reaching the integral limit doesn't result in a loop fault.

Bit 9 – Excess Following Error

If the following error equals or exceeds the maximum following error programmed into the parameter block, the module turns this bit on and activates OUTPUT 2 if configured as the loop fault output.

Bit 10 – PID Error

If the positioning error equals or exceeds the maximum PID error programmed into the parameter block (if the PID bit is on), the module turns this bit on and activates OUTPUT 2 if configured as the loop fault output.

Bit 11 – Immediate Stop

The stop bit turns on when the module recognizes a hardware stop input or immediate stop command. The module also activates OUTPUT 2, if configured as the loop fault output, when it performs an immediate stop.

Bit 12 – Discrete Input Fault

The discrete input fault bit turns on when the module detects a fault in the discrete input circuitry. In this event, the module also activates OUTPUT 2 if configured as the loop fault output. The following conditions will cause a discrete input fault:

- loss of discrete input power
- discrete input circuitry fault

Bit 13 – Feedback Fault

The feedback fault bit turns on when the module detects a fault in the transducer interface circuitry. In this event, the module also activates OUTPUT 2 if configured as the loop fault output. The following conditions will cause a feedback fault:

- loss of transducer power
- internal loop-back fault
- excessive change in velocity
- loss of feedback
- position exceeds maximum transducer length

Bit 14 – Analog Fault

The analog fault bit turns on when the module detects a fault in the analog circuitry. In this event, the module also activates OUTPUT 2 if configured as the loop fault output. The following conditions will cause an analog fault:

- loss of analog power
- analog power supply voltage out of tolerance
- analog circuitry fault

Bit 15 – Internal Fault

The internal fault bit turns on if the module detects a fault in the circuitry powered by the backplane. In this event, the module also activates OUTPUT 2 if configured as the loop fault output. If this fault occurs, return the module to your Allen-Bradley representative.

Position/Error/Diagnostic Words

You can use these words to display diagnostic information, current axis position, or the following error. You select the information to be displayed through bits in the command block. You can also view all three parameters simultaneously by specifying diagnostics for words 4, 5 and 8, 9; position information for words 12, 13 and 14, 15; and following error for words 16, 17 and 18, 19. These selections use the extended status information. See Chapter 7 for details on the command block.

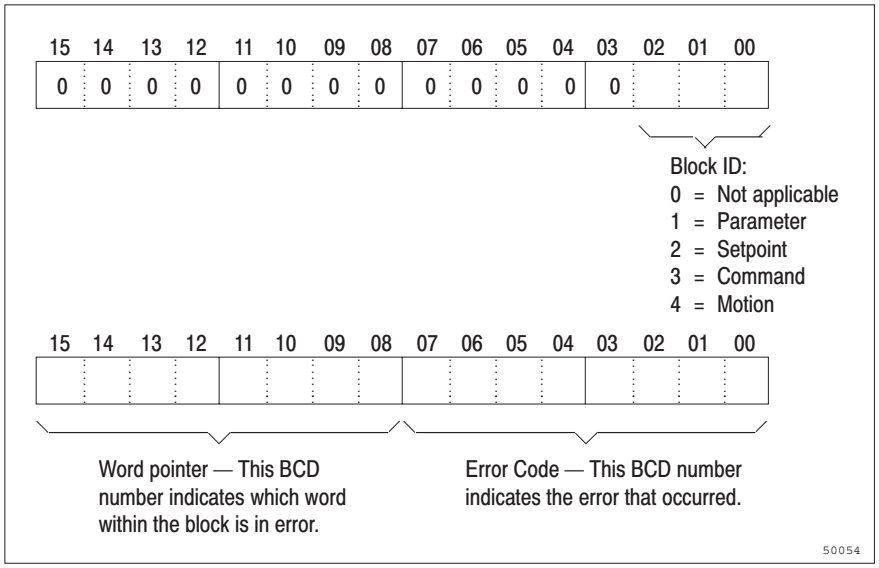
Diagnostic Information (words 4, 5 and 8, 9)

After a reset command or powerup, the module displays diagnostic information so you can detect parameter block errors. (The module doesn't accept command blocks until after it receives a valid parameter block.)

Use the diagnostic words to determine the cause of a block transfer error. The block ID identifies the last block received by the module. It is updated with each block transfer. The word pointer identifies the location of the problem and the error code determines the nature of the problem.

The diagnostic information is displayed in BCD format. See Appendix H for an explanation of numbering formats.

Figure 6.5
Diagnostic Format


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0 0 0 0 0 0
Block ID:
0 = Not applicable
1 = Parameter
2 = Setpoint
3 = Command
4 = Motion
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Word pointer — This BCD number indicates which word within the block is in error.
Error Code — This BCD number indicates the error that occurred.
50054

The second of the two diagnostic words gives the error code and points to the word where the error occurred. Table 6.A shows the error codes.

Table 6.A
Error Codes

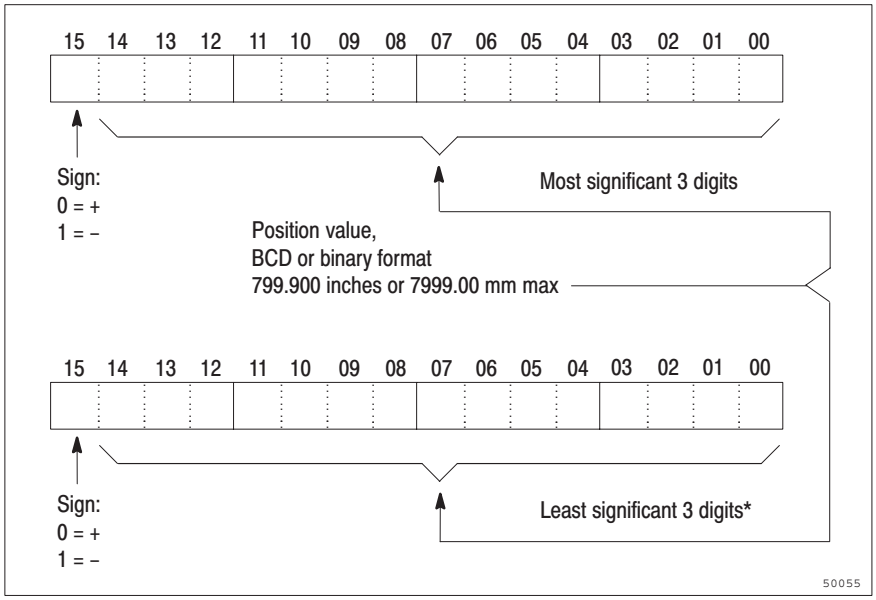
Code	Definition
00	No errors detected
01	Invalid block identifier
02	Non-BCD number entered
03	Invalid bit setting, unused bits must be set to zero
04	Data is out of range
05	Invalid number of axes programmed
06	Setpoint is not defined
07	Setpoint commanded while in manual mode
08	Position exceeds a software travel limit
09	Attempted to switch to auto mode with axis in motion
10	Attempted to switch to manual mode with axis in motion
11	Velocity exceeds maximum
12	High jog rate > maximum velocity
13	Low jog rate > high jog rate
14	Maximum PID error must be outside the PID band
15	Incorrect block length
16	First block after powerup must be a parameter block
17	Negative travel limit $\geq$ positive travel limit
18	Jog commanded while in auto mode
19	Forward and reverse jogs commanded simultaneously
20	Block transfer write attempted before module confirmed all power on wiring arm
21	Specified velocity exceeds maximum velocity for direction of motion
22	Motion segment ID not defined
23	Motion segment commanded while in manual mode
24	A motion segment is attempting to use an output which is not configured as a programmable output
25	Motion segment ID previously defined in same motion block

Position Information (words 4, 5; 8, 9; 12, 13; and 14, 15)

The position words give the present position measured at the transducer. The position information is in either BCD or binary format. You choose the format you want through the parameter block. Binary format is compatible with integer data (16-bit 2's complement) used by PLC-5 family programmable controllers. See Appendix H for an explanation of numbering formats.

The maximum position displayed is ±799.900 inches or ±7999.00 millimeters. If the status words 4, 5 and 8, 9 are specified in the command block to display position, when the axis exceeds the maximum, the maximum is displayed and the position valid bit in the second status word turns off.

Figure 6.6
Position Format


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant 3 digits
Position value,
BCD or binary format
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits*
50055

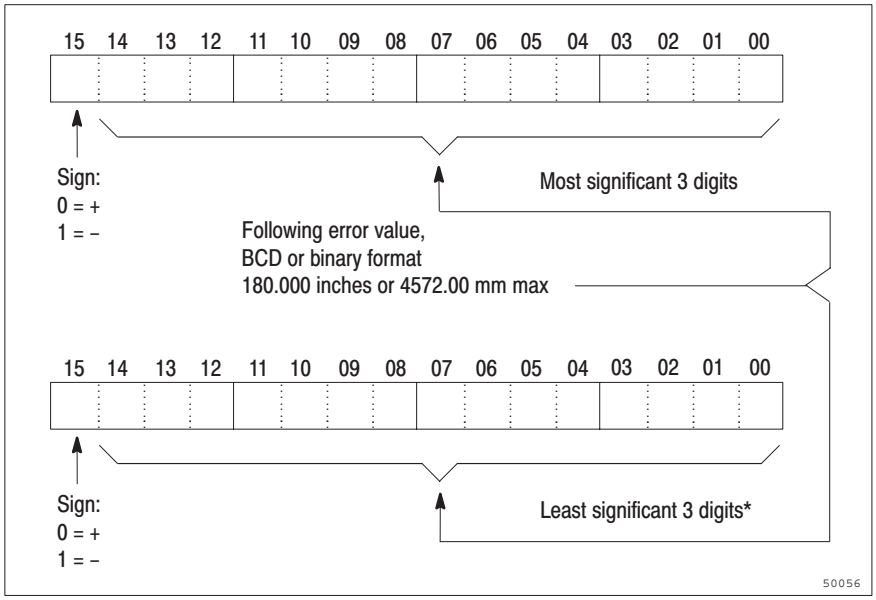
* As long as the parameter control word (see Chapter 7) is configured for binary (bit 3=0) and single word (bit 7=1) formats, values between -32.768 inches and 32.767 inches (-327.68 mm and 327.67 mm) may be displayed entirely in the second word. The first word will be zero.

Following Error Information (words 4, 5; 8, 9; 16, 17; and 18, 19)

Following error is determined by subtracting the actual position of the axis measured at the transducer from the desired position calculated by the module. The desired position is calculated every two milliseconds based on the acceleration, deceleration and velocity of the move. The error information is in either BCD or binary format. You choose the format you want through the parameter block. Binary format is compatible with integer data (16-bit 2's complement) used by PLC-5 family programmable controllers. See Appendix H for an explanation of numbering formats.

The maximum following error displayed in the status block is ±180.000 inches or ±4572.00 millimeters. If the status words 4, 5 and 8, 9 are specified in the command block to display errors, when the axis exceeds the maximum, the maximum is displayed and the error valid bit in the second status word turns off.

Figure 6.7
Following Error Format

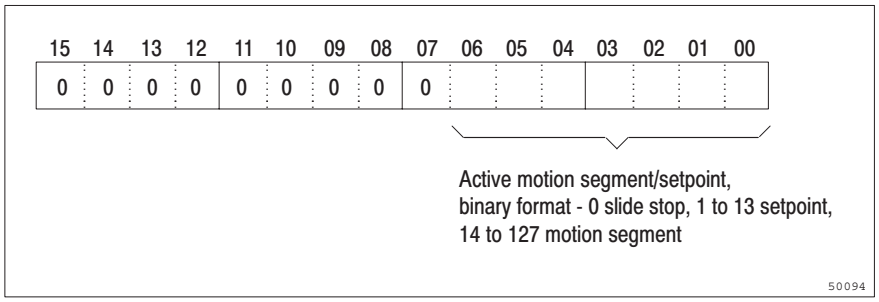

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant 3 digits
Following error value,
BCD or binary format
180.000 inches or 4572.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits*
50056

* As long as the parameter control word (see Chapter 7) is configured for binary (bit 3=0) and single word (bit 7=1) formats, values between -32.768 inches and 32.767 inches (-327.68 mm and 327.67 mm) may be displayed entirely in the second word. The first word will be zero.

Active Motion Segment/Setpoint (words 10 and 11)

Words 10 and 11 of the extended status block contain the active motion segment or setpoint number.

Figure 6.8
Active Motion Segment/Setpoint


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0 0
Active motion segment/setpoint,
binary format - 0 slide stop, 1 to 13 setpoint,
14 to 127 motion segment
50094

Measured Velocity (words 20 and 21)

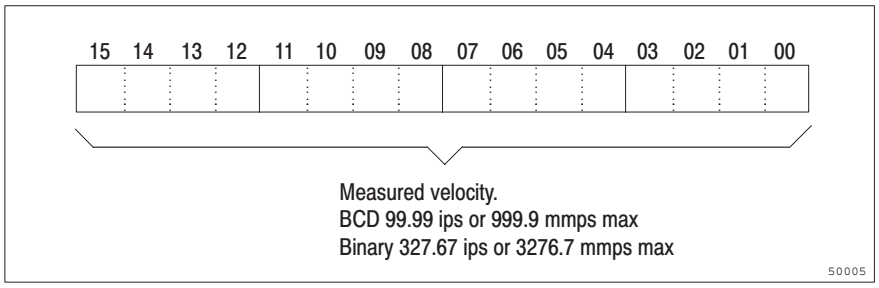
Measured velocity is the instantaneous speed of the axis measured at the transducer. This velocity is calculated using a moving average over the previous 20, 50 or 100 milliseconds (depending on the velocity commanded for the move). For slow moves, a 100 millisecond averaging interval is used to improve resolution. For fast moves, a 50 or 20 millisecond averaging interval is used to improve responsiveness.

Measured velocity is always positive, regardless of the direction of travel.

Table 6.B
Averaging Interval for Various Commanded Velocities

Commanded Velocity		Averaging Interval
ips	mmps	
0.00 - 10.00	0 - 254.0	100 ms
10.01 - 20.00	254.1 - 508.0	50 ms
> 20.00	> 508.0	20 ms

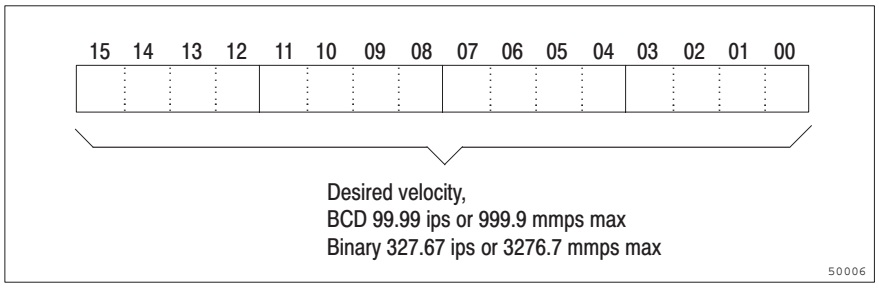
Figure 6.9
Measured Velocity Format



Desired Velocity (words 22 and 23)

The module calculates the desired velocity once every two milliseconds based on the acceleration, deceleration and velocity specified for the move. The desired velocity is a theoretical number representing the speed that the module wishes to achieve, and not necessarily the actual velocity of the axis. The desired velocity is always positive, regardless of the direction of travel.

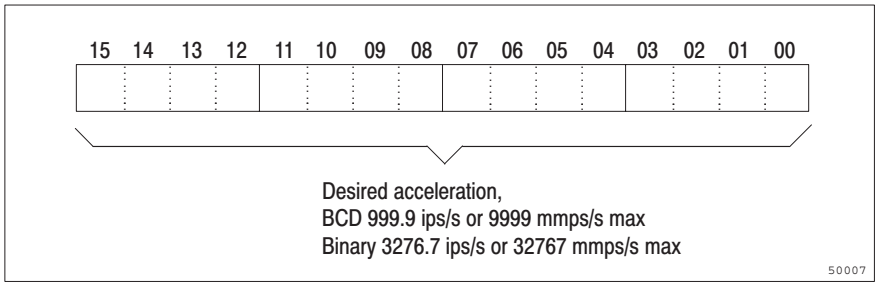
Figure 6.10
Desired Velocity Format



Desired Acceleration (words 24 and 25)

The module calculates the desired acceleration once every two milliseconds, based on the velocity smoothing constant and maximum acceleration specified for the move. The desired acceleration is a theoretical number representing the rate of velocity increase that the module wishes to achieve, and not necessarily the actual rate of acceleration achieved.

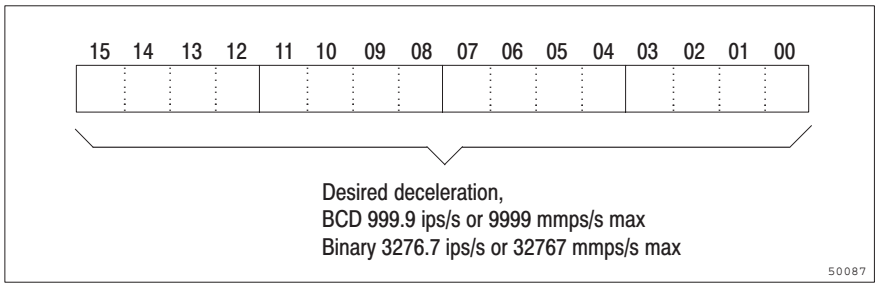
Figure 6.11
Desired Acceleration Format


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Desired acceleration,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50007

Desired Deceleration (words 26 and 27)

The module calculates the desired deceleration once every two milliseconds based on the velocity smoothing constant and maximum deceleration specified for the move. The desired deceleration is a theoretical number representing the rate of velocity decrease that the module wishes to achieve, and not necessarily the actual deceleration achieved.

Figure 6.12
Desired Deceleration Format


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Desired deceleration,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50087

Percent Analog Output (words 28 and 29)

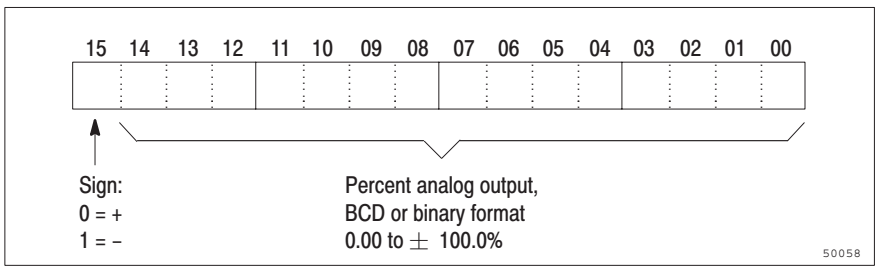
Analog output is controlled by the module's PID and feedforward control algorithms. It represents the percentage of the full scale analog output used to control the servo valve. The maximum full-scale output is determined by the hardware switches (see Chapter 5) and the analog range word (see Chapter 7).

Example: If the analog output switches are configured for ±100 mA, and an analog range of 50% is specified in the parameter block, an analog output of +100.0% represents +50 mA and –100.0% represents –50mA with respect to the +ANALOG output. If the most significant bit of the analog range word is set to reverse the analog output polarity, +100% will still represent +50 mA with respect to the +ANALOG output

The percent analog output is updated even when the analog outputs are disabled by a fault or by the parameter control word.

The percent analog output can be used to monitor the output required to keep the axis stationary. If a large value is detected (above 15%), the servo valve may be out-of-null, or the integral term of the PID algorithm may have driven the analog output towards the minimum or maximum (i.e., integral windup). You can limit integral windup by setting the integral term limit (see Chapter 7) to 10 or 15%.

Figure 6.13
Percent Analog Output



Maximum Velocity (words 30, 31 and 32, 33)

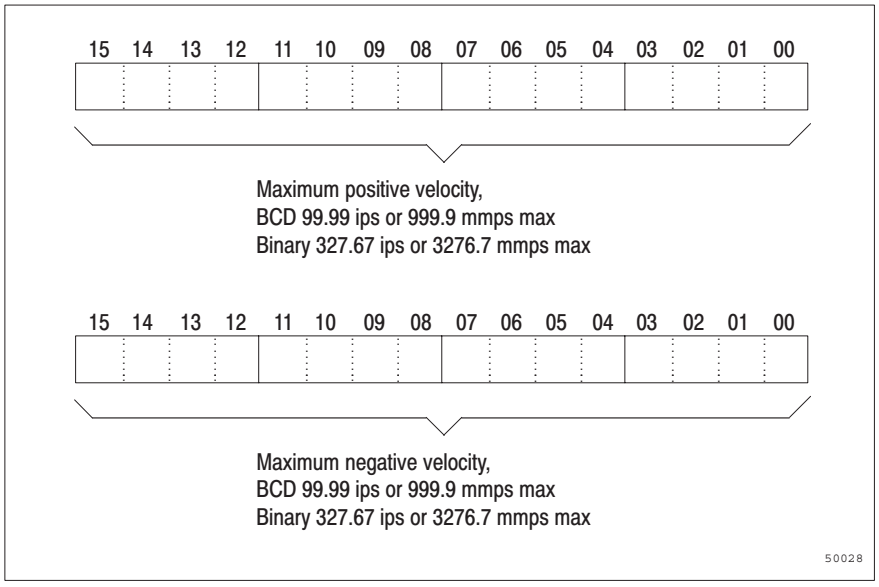
The maximum velocity words represent the maximum speed that the system is capable of moving in each direction, and not necessarily the maximum velocity of a particular move.

The module calculates the theoretical maximum positive and negative velocities by monitoring jogs or setpoint or motion block moves and extrapolating the maximum speeds possible with the servo valve fully open.

The maximum velocity values returned by the module can greatly simplify the tuning procedures for your axes. You can enter the maximum positive velocity as the optimal positive analog calibration constant, and the maximum negative velocity as the optimal negative analog calibration constant. The module will use these values to adjust the PID and feedforward gains for directional differences in system performance.

The maximum velocity words can also be used to monitor the performance of the hydraulics. If the maximum velocity changes dramatically, the hydraulics may require servicing.

Figure 6.14
Maximum Velocity Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum positive velocity,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum negative velocity,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
50028

While every effort has been made to ensure that the maximum velocity calculations are foolproof, the following limitations do exist:

- the module will ignore moves where a constant velocity is not achieved. The maximum velocity calculations are only accurate when the axis stabilizes at a constant velocity.

- the accuracy is degraded if the axis is unstable or if the velocity is extremely low. Velocities at or above 10% of the maximum velocity work best.
- the maximum velocities calculated by the module will not be accurate if motion is impeded by physical obstructions.
- the maximum velocity predictions will vary slightly for moves at different velocities due to non-linearities in the hydraulic system. If it is critical that the module perform best at a particular velocity, then that velocity should be used to determine the optimal analog calibration constants. Otherwise, it is best to use a moderately low velocity (10% of the maximum velocity) to optimize the performance near the setpoint.

Chapter 7

Formatting Module Data (WRITES)

Data Blocks Used in Write Operations

Data blocks that you set up in the PLC data table enable you to control the module from your PLC programs. There are four types of data blocks used in write operations. The three discussed in this chapter are parameter, setpoint and command blocks. The motion block is discussed in Chapter 9.

Parameter Block (Required)

The parameter block contains loop configuration information. The module must receive and acknowledge the parameter block before it can receive setpoint, motion and command blocks. You will normally only send a parameter block to the module after reset or powerup. If you do send one during module operation, the module will not activate the new parameters until axis motion stops.

Setpoint Block (Optional)

By sending a setpoint block, you can specify up to 12 setpoints for each axis. You can move to a selected setpoint by sending a command block.

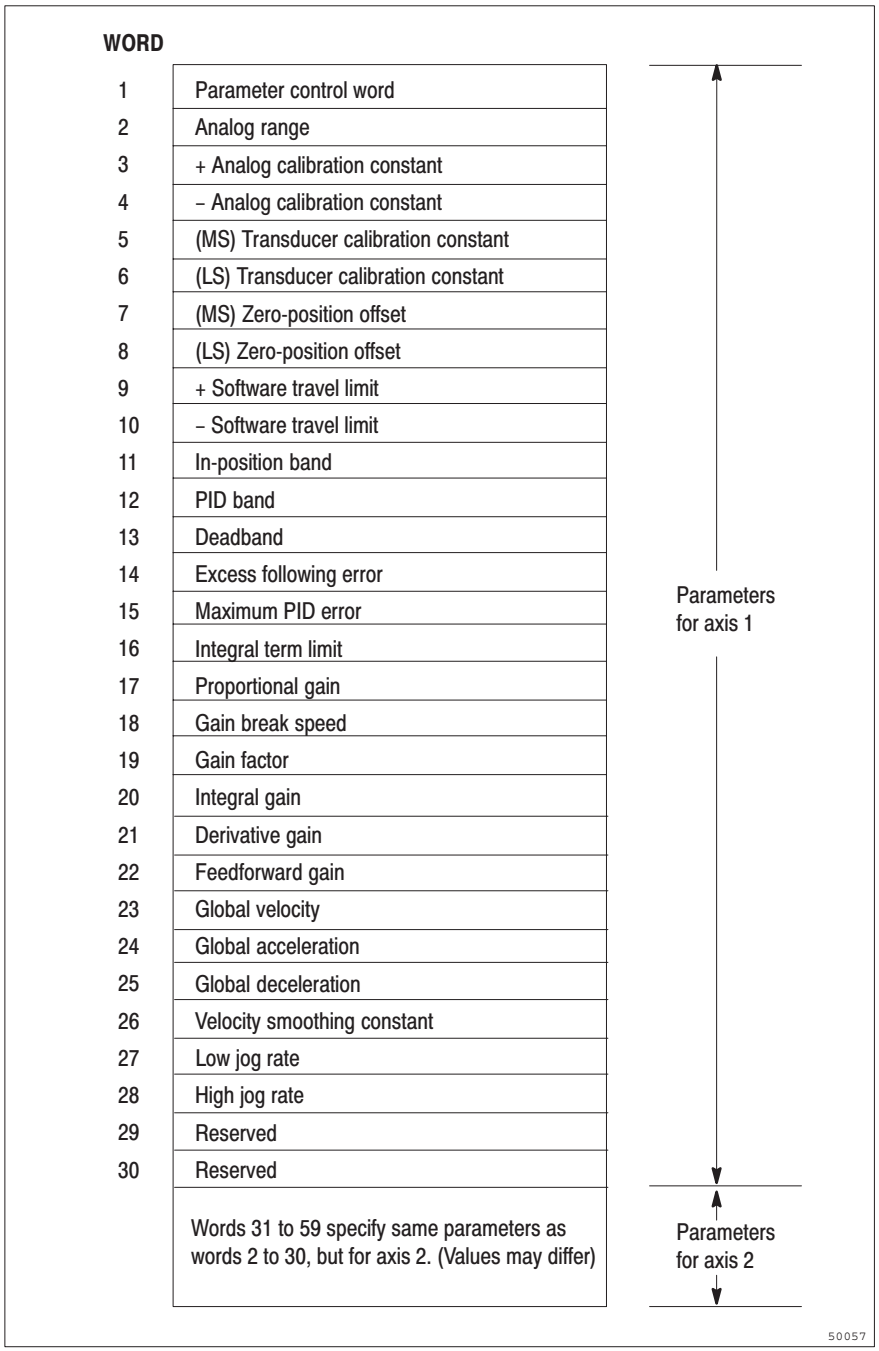
Command Block (Required)

By sending a command block, you begin the movement of one axis or both axes simultaneously. This requires a jog command in manual mode or either a setpoint or motion segment move command in auto mode.

Parameter Block

The parameter block contains parameters to configure the two axes controlled by the module. Figure 7.1 shows parameter block word assignments.

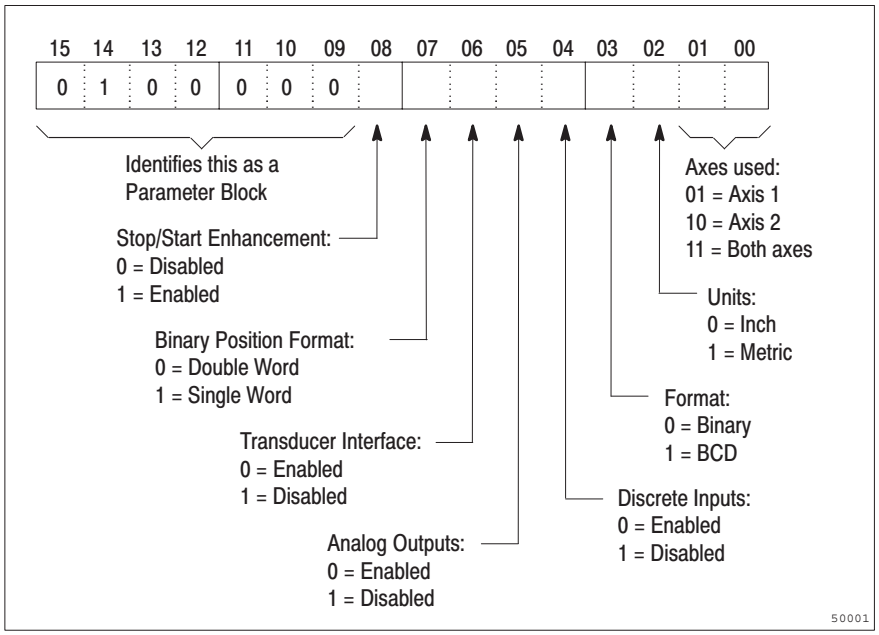
Figure 7.1
Parameter Block Word Assignments



Parameter Control Word (word 1)

The parameter control word identifies the block as a parameter block and provides configuration information common to both loops. You can also disable the transducer interface, analog outputs, and discrete inputs by setting the appropriate bits. If all three sections are disabled, you can test the programmable controller program without connecting the wiring arm to the module. Unused sections do not have to be powered.

Figure 7.2
Parameter Block Control Word



Bits 0 and 1 – Axes Used

Bits 0 and 1 determine which axes are controlled by the module. You can use either one separately, or both. The module performs error processing independently for each axis. If it detects a format error for one axis, it discards all new parameters for that axis.

Bit 2 – Inch/metric

Bit 2 selects between metric and imperial units.

Bit 3 – Binary/BCD

Bit 3 determines the format of the data contained in block transfer reads and writes. BCD format provides compatibility with older programmable controllers. Binary format provides compatibility with the PLC-5, which uses integer (16-bit 2's complement) data.

Bit 4 – Discrete Inputs

Setting this bit to 1 disables the discrete inputs. The state of the inputs can still be monitored by the status block, but the function of each input is disabled. Discrete faults are reported in the status block, but OUTPUT 2, if configured as the loop fault output, is not activated when a discrete input fault occurs. The discrete input section does not have to be powered when the inputs are disabled. If disabled, the function of each input is as follows:

Discrete Input:	Function:
stop input	disabled
auto/manual input	auto
jog inputs	disabled
start input	disabled

Bit 5 – Analog Outputs

Setting this bit to 1 disables the analog outputs by opening internal relays. Analog faults are still reported in the status block, but OUTPUT 2, if configured as the loop fault output, is not activated when an analog fault occurs. The analog section does not need to be powered when the analog outputs are disabled.

The percent analog output is displayed in the status block even if the analog outputs are disabled. This allows you to test the programmable controller program.

Bit 6 – Transducer Interface

Setting this bit to 1 disables the transducer interface. The interrogate pulse is still sent and feedback faults are reported, but the transducer reading is ignored. The transducer section does not need to be powered when the transducer interface is disabled.

When the transducer interface is disabled the module will simulate transducer feedback to help you test the programmable controller program. The position changes at the programmed acceleration, velocity and deceleration when a setpoint, motion segment or jog command is issued. The following error will remain zero.

Bit 7 – Binary Position Format

When bit 7 is set to 1, and binary format is specified in the parameter control word (bit 3 = 0), the module can display position and error values between -32.768 and 32.767 inches (-327.68 and 327.67 mm) in the second word of the position or following error in the status block. This feature allows applications with a stroke of less than 32 inches to monitor position and error with a single integer. If the position or error exceeds the maximum, the module automatically reverts to double word format.

Setting this bit to 0, or selecting BCD format in the parameter control word bit (bit 3 = 1), configures the module to display position and error in double word format. The first word displays inches or centimeters and the second word displays fractions of an inch or centimeter. See Table 7.A.

Table 7.A
Single and Double Word Format Representations

Position/Error	Double Word Format		Single Word Format	
	First Word	Second Word	First Word	Second Word
+6.000 inches	6	0	0	6000
–32.768 inches	–32	–768	0	–32768
327.67 mm	32	767	0	32767
–10.00 mm	–1	0	0	–1000

Bit 8 - Stop/Start Enhancement

When this bit is set, it causes a positive rising edge hardware start input to be accepted during axis motion, similar to the software start bit in the command block. Also, as long as the software slide stop bit in the command block is high, the axis remains stationary since no setpoint (or motion segment if a motion block is being used), can be initiated. While most new applications can set this bit, existing applications may clear it to ensure backwards compatibility.

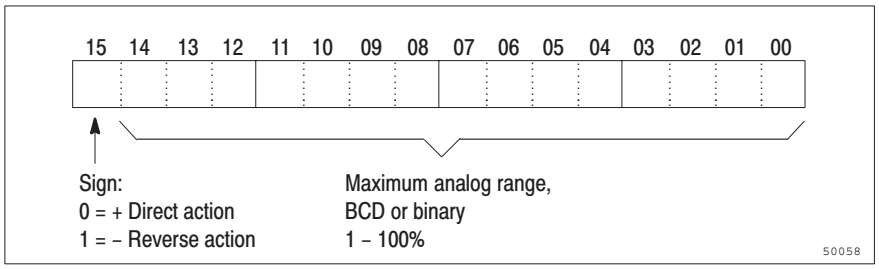
Analog Range (words 2 and 31)

The analog range parameter specifies the maximum analog output available for commanding motion. It may be positive or negative. Analog range is a percentage of the range selected through the analog output DIP switches. (See Chapter 5.) For example, if the analog range is specified as +100%, the direct action analog output ranges from -10 V to 10 V, -20 mA to 20 mA, -50 mA to 50 mA, or -100 mA to 100 mA, depending on the setting of the analog output switches. If the analog range is specified as -100%, the reverse action output ranges from 10 V to -10 V, 20 mA to -20 mA, 50 mA to -50 mA or 100 mA to -100 mA. Use this parameter to make sure that the module does not exceed the maximum rating of the external device.

Important: If the maximum analog range is negative, the +ANALOG and –ANALOG outputs behave as if they were physically reversed.

ATTENTION: An incorrect sign for the analog range can cause the axis to accelerate out of position when you close the loop.

Figure 7.3
Analog Range Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = + Direct action
1 = – Reverse action
Maximum analog range,
BCD or binary
1 – 100%
50058

ATTENTION: Make sure that the analog output doesn't exceed the maximum for your device.

Example: To set an analog output range of ±70 mA:

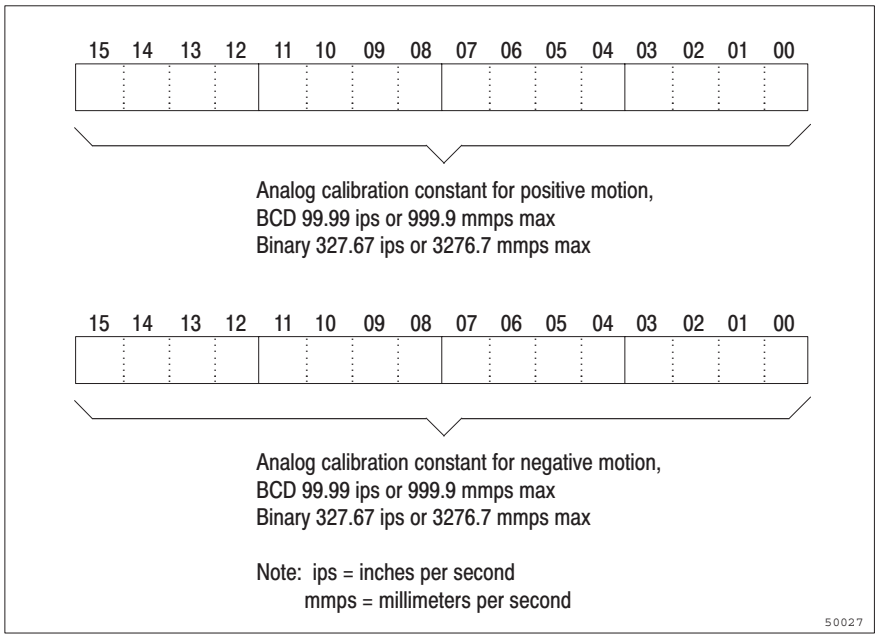
- configure the analog output DIP switches for ±100 mA
- specify an analog range of 70% in the analog range word

Analog Calibration Constants (words 3, 4 and 32, 33)

The analog calibration constants specify the highest velocity that the axis can attain in each direction. These rates, associated with the maximum positive and negative analog outputs, are 327.67 ips (inches per second) or 3276.7 mmps (millimeters per second) for binary format. For BCD the maximum is 99.99 ips or 999.9 mmps.

The module uses these parameters to determine the relationship between the analog output and the speed of the axis. A separate parameter for each direction compensates for directional differences of the device (the zero-position offset defines the positive and negative directions). The module performs this compensation by adjusting the loop gains (proportional, integral, derivative, and feedforward).

Figure 7.4
Analog Calibration Constant Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Analog calibration constant for positive motion,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Analog calibration constant for negative motion,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
Note: ips = inches per second
mmps = millimeters per second
50027

For servo valves, the analog calibration constants can be roughly calculated from the diameter of the cylinder and the maximum flow rate of the valve. You will fine-tune these parameters when you perform the tuning procedure given in Chapter 8.

Transducer Calibration Constant (words 5, 6 and 34, 35)

The module uses the transducer calibration constant to convert the transducer generated pulse width into an axis position reading.

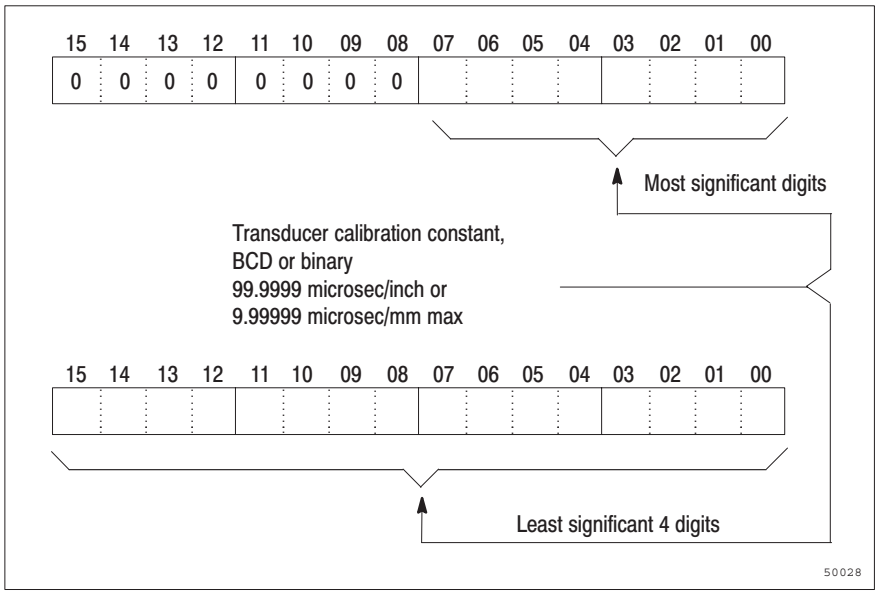
Calculate the transducer calibration constant by multiplying the figure stamped on the side of the transducer head by the number of circulations that you are using. This figure varies slightly from one transducer to another. It is typically 9.0500 microseconds per inch or 0.35600 microseconds per millimeter.

Example: If your transducer is stamped with 9.0500 microseconds per inch and you've programmed your digital interface box for four circulations, your transducer calibration constant would be:

4 x 9.0500 = 36.2000

See Chapter 4 to determine the optimum number of circulations for your system and Chapter 8 for a procedure for verifying the transducer calibration constant.

Figure 7.5
Transducer Calibration Constant Words

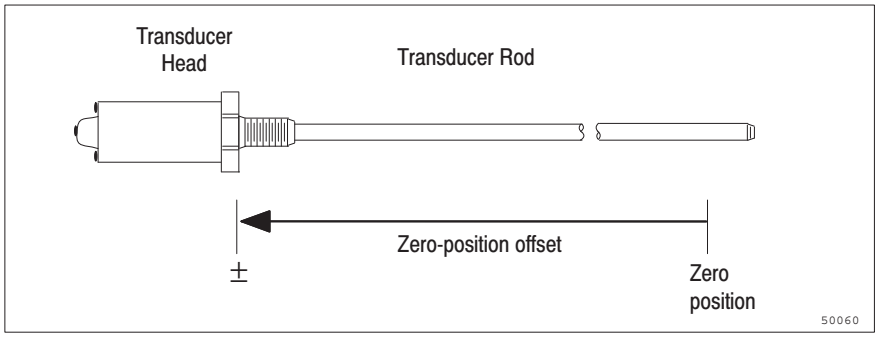

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0
Most significant digits
Transducer calibration constant,
BCD or binary
99.9999 microsec/inch or
9.99999 microsec/mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Least significant 4 digits
50028

Zero-Position Offset (words 7, 8 and 36, 37)

The zero position offset words define the origin of the coordinate system. Zero-position can be located within or outside the transducer's active range. This allows positions to be measured relative to locations outside the range of axis motion. The software travel limits and setpoint positions must reside within the transducer's active range.

Important: The active range of the transducer is halved by each increase in the number of circulations of the digital interface box.

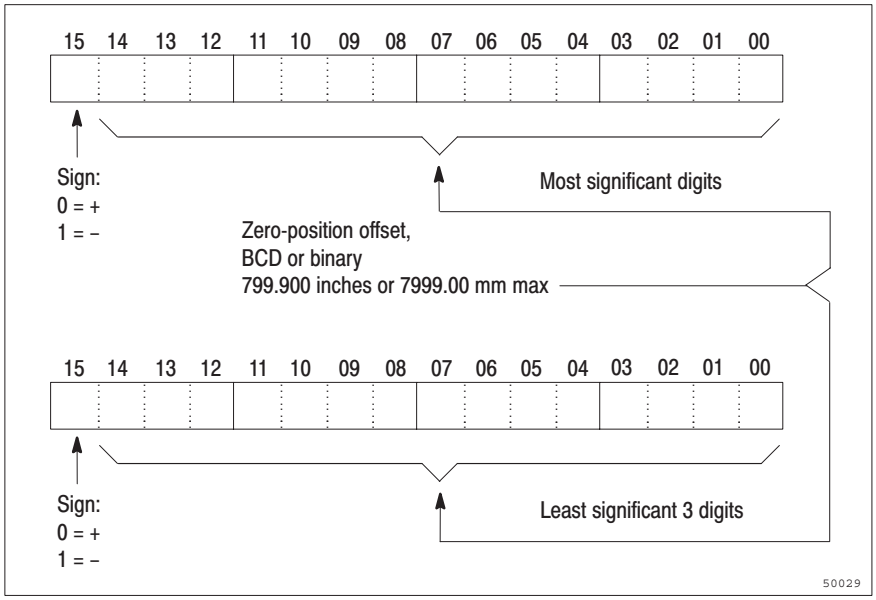
Figure 7.6
Zero-Position Offset



Important: If you change the axis polarity, exchange the forward and reverse analog calibration constants. The zero-position offset defines the direction of forward and reverse motion.

Calculate the zero-position offset by measuring the distance between the zero-position and the transducer's head, as shown above. The module accepts a maximum of ±799.900 inches (±7999.00 millimeters). The sign defines whether the transducer head is on the positive or negative end of the axis. If the zero-position offset is zero, then the positive direction is away from the transducer head, in the extend direction.

Figure 7.7
Zero-Position Offset Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Zero-position offset,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50029

Important: If you select binary format, both words are represented as 2's-complement integers compatible with the PLC-5. See Appendix H for examples of these words.

Software Travel Limits (words 9, 10 and 38, 39)

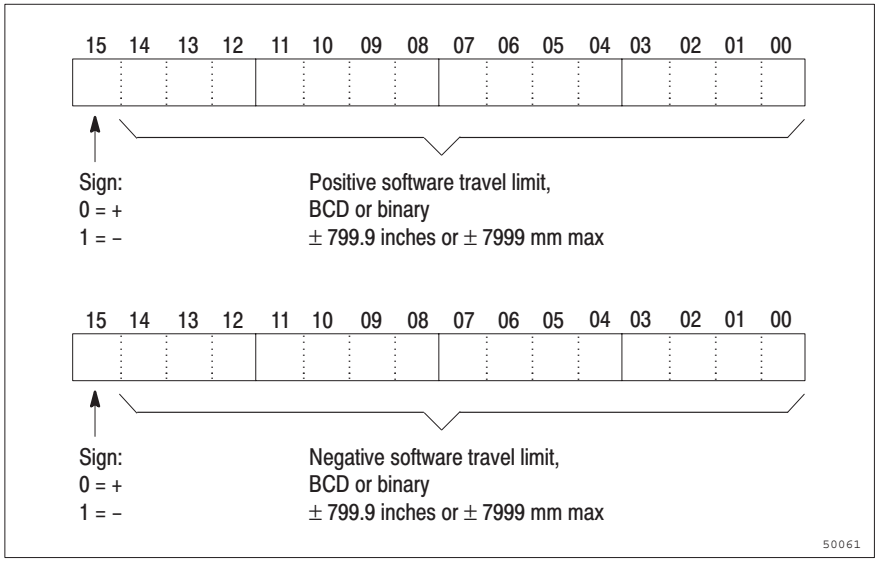
When a setpoint command calls for the axis to move beyond a software travel limit, the module aborts the move and reports a programming fault. When a jog command calls for the axis to move beyond a software travel limit, axis movement will decelerate and stop at the limit.

The software travel limits must be within the active range of the transducer. The active range of your transducer is halved by each increase in the number of circulations of your digital interface box.

If you program both software travel limits to zero, the module defaults to a negative software travel limit of 0 and a maximum positive software travel limit that is 180.0 inches or 4572 mm for one recirculation. If you select binary format, the software travel limits are represented as 2's-complement integers.

ATTENTION: To guard against equipment damage, we recommend that you set software travel limits to match your axis length.

Figure 7.8
Software Travel Limit Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Positive software travel limit,
BCD or binary
± 799.9 inches or ± 7999 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Negative software travel limit,
BCD or binary
± 799.9 inches or ± 7999 mm max
50061

Zero-Position and Software Travel Limit Examples

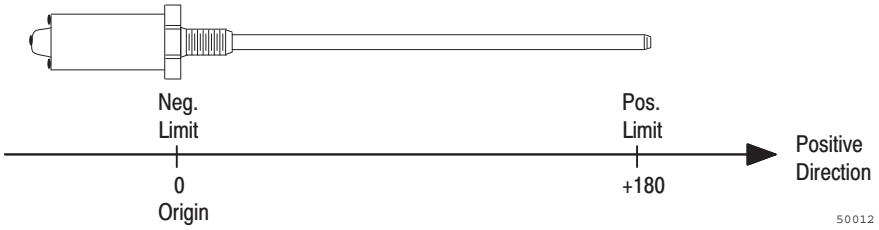
The zero position offset and software travel limits can be difficult to understand so the following examples have been provided. Note that the examples show zero position and software travel limits relative to the movement of the magnet along the transducer. The actual movement of a workpiece depends on how the transducer is mounted in a given system.

Example: Default Configuration

If the zero-position and software travel limits are 0, all measurements are relative to the transducer head and the positive direction is towards the end of the transducer. If you program both software travel limits to 0, the module defaults to the maximum and minimum that it can measure. In this example, the negative limit is at the origin and the positive limit is at the maximum distance that the module can measure: 180 inches for one circulation.

Figure 7.9
Default Configuration

Zero-position = 0.000
Positive Limit = 0.0
Negative Limit = 0.0

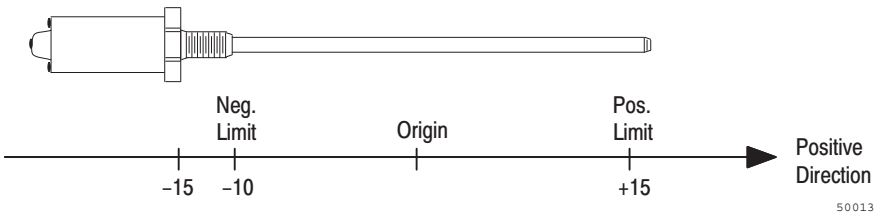


Example: Extending in the Positive Direction

In this example, the transducer head is -15 inches from the origin. Notice that all measurements are relative to the origin. The *value* of the zero position offset determines the distance between the origin and the transducer head. The *sign* of the zero position offset indicates that the transducer head is in the negative direction.

Figure 7.10
Extending in the Positive Direction

Zero-position = –15.000
Positive Limit = +15.0
Negative Limit = –10.0

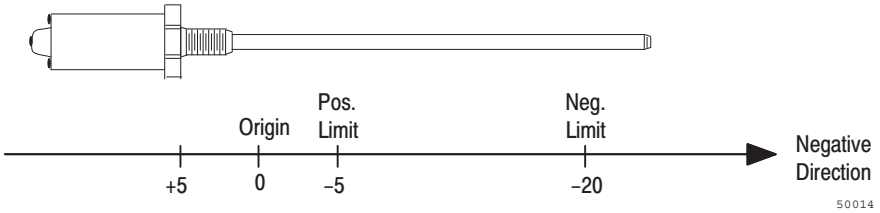


Example: Retracting in the Positive Direction

In this example, the polarity of the axis has been reversed. The positive direction is now towards the transducer head as indicated by the sign of the zero position offset. Notice that the software travel limit in the positive direction can have a negative sign.

Figure 7.11
Retracting in the Positive Direction

Zero-position = +5.000
Positive Limit = –5.0
Negative Limit = –20.0

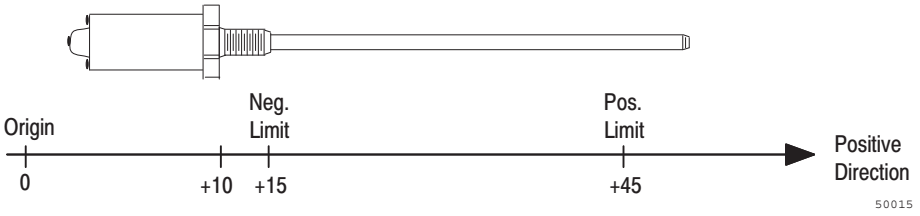


Examples: Zero-Position Left of the Transducer Head

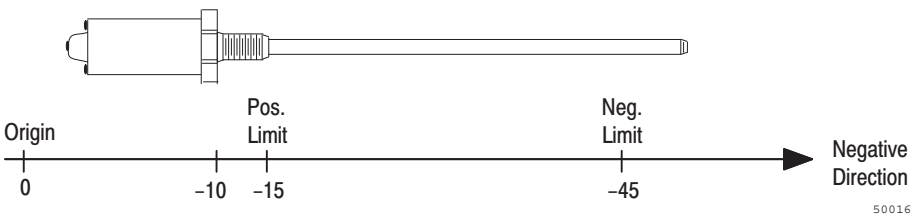
The next two examples demonstrate configurations with the origin past the fully retracted position.

Figure 7.12
Zero-Position Left of the Transducer Head

Zero-position = +10.000
Positive Limit = +45.0
Negative Limit = +15.0



Zero-position = –10.000
Positive Limit = –15.0
Negative Limit = –45.0

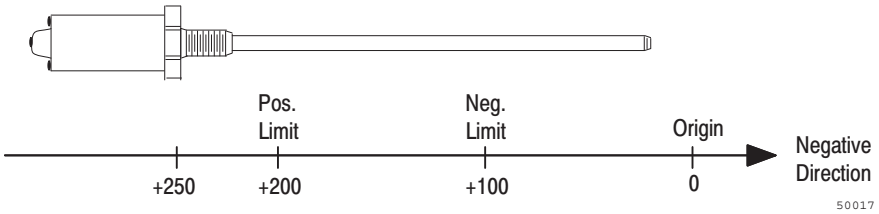


Examples: Zero-Position Past the End of the Transducer

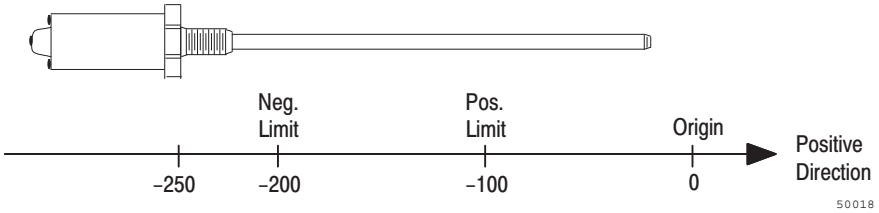
The next two examples show the origin past the fully extended position.

Figure 7.13
Zero-Position Past the End of the Transducer Head

Zero-position = +250.000
Positive Limit = +200.0
Negative Limit = +100.0



Zero-position = −250.000
Positive Limit = −100.0
Negative Limit = −200.0

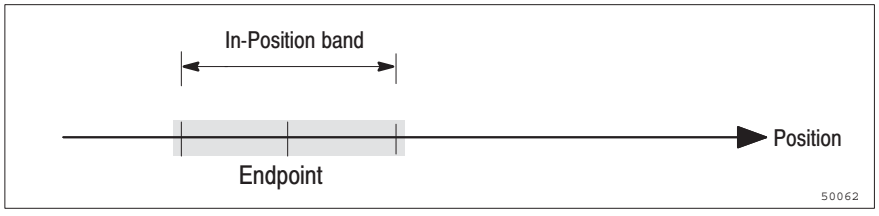


In-Position Band (words 11 and 40)

The in-position band is the area around an endpoint where the in-position bit turns on. An endpoint can be the result of a setpoint or motion segment move or a jog. The axis is in-position if:

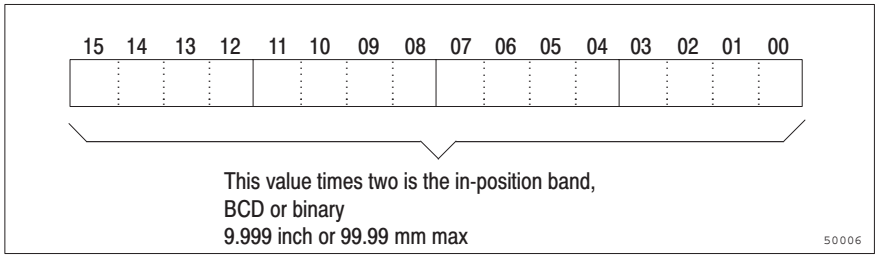
- the axis feed is complete (i.e., desired velocity is zero)
- the following error has closed to within the in-position band

Figure 7.14
In-Position Band



If you leave the in-position band undefined (at zero), the module automatically defaults to twice the value of the position resolution. For one circulation, this would be 0.004 inches.

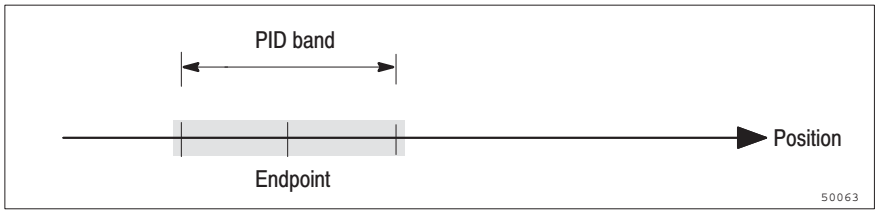
Figure 7.15
In-Position Band Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
This value times two is the in-position band,
BCD or binary
9.999 inch or 99.99 mm max
50006

PID Band (words 12 and 41)

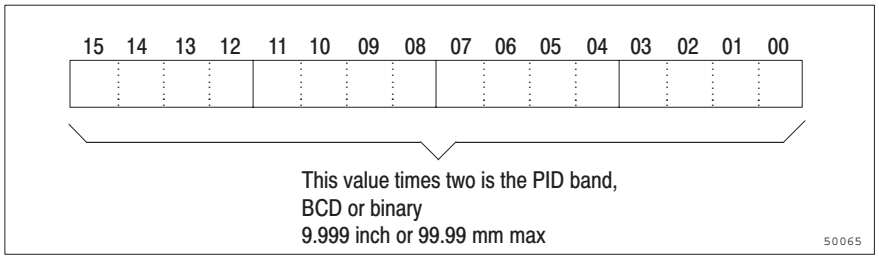
If the axis is within the PID band and the desired velocity is zero, the module enables the integral and derivative components for final positioning of the axis. (See Chapter 2.) If the PID band is programmed to zero, the integral and derivative terms remain disabled.

Figure 7.16
PID Band



The maximum value of the PID band word is 9.999 inches or 99.99 mm.

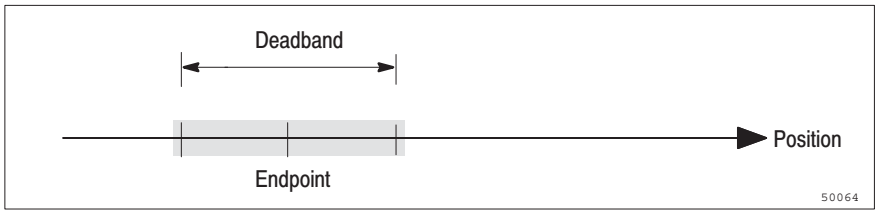
Figure 7.17
PID Band Word



Deadband (words 13 and 42)

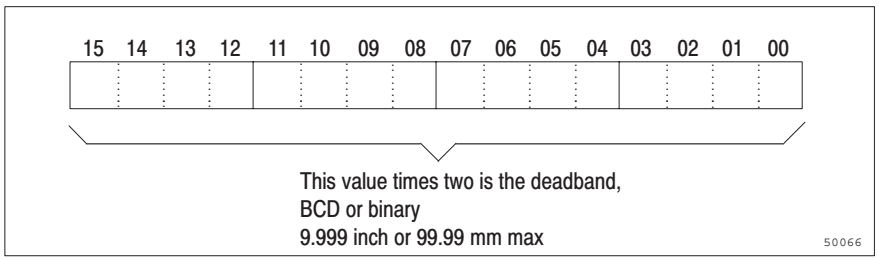
The deadband parameter lets you select an error range on either side of a commanded endpoint where the integral term of the PID algorithm doesn't change.

Figure 7.18
Deadband



The module uses the deadband only after the axis crosses the endpoint. The deadband helps reduce oscillations around the endpoint.

Figure 7.19
Deadband Word

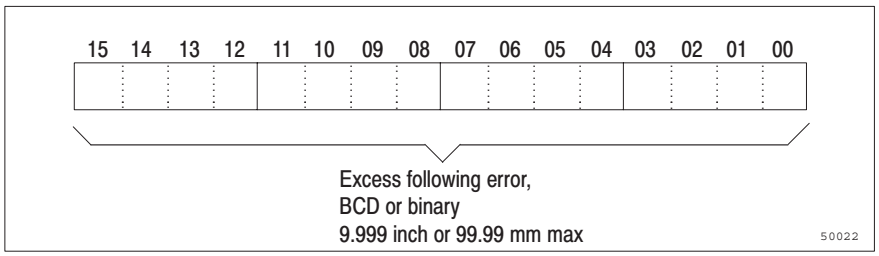


Excess Following Error (words 14 and 43)

The excess following error is the maximum allowable axis error above the expected following error at the programmed velocity for the current move. The expected following error for a given velocity equals the velocity divided by the proportional gain.

When the following error reaches the maximum value permitted, as specified by the excess following error parameter, the module initiates an immediate stop (loop fault). To disable excess following error checking, specify an excess following error of zero.

Figure 7.20
Excess Following Error Word



Example: If axis movement is 5 ips and proportional gain (K_P) at that speed is 0.05 ips/mil (where 1 mil = .001 inch), then

Expected Following Error = (5)/(0.05) = 100 mil

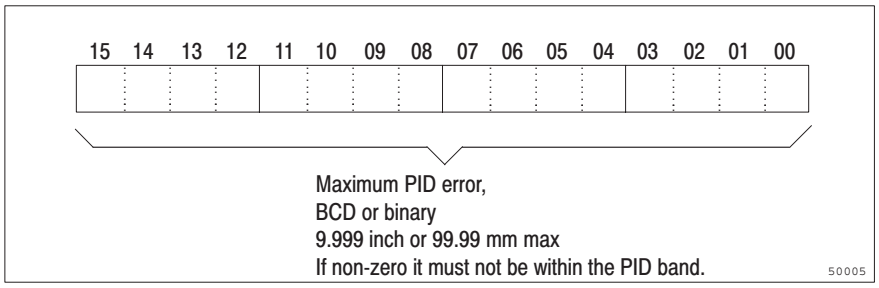
If you specify an excess following error of 50 mil, then an immediate stop will occur if the following error reaches 150 mil (the expected following error plus 50 mil).

Maximum PID Error (words 15 and 44)

The maximum PID error is the maximum position error when the integral and derivative components are enabled for final axis positioning (i.e., when the desired velocity is zero and the axis is within the PID band).

When the maximum PID error is exceeded, the module initiates an immediate stop (loop fault).

Figure 7.21
Maximum PID Error Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum PID error,
BCD or binary
9.999 inch or 99.99 mm max
If non-zero it must not be within the PID band.
50005

The maximum value of this word is 9.999 inches or 99.99 mm. The maximum PID error must not be within the PID band unless the PID error checking is disabled. To disable PID error checking, specify zero.

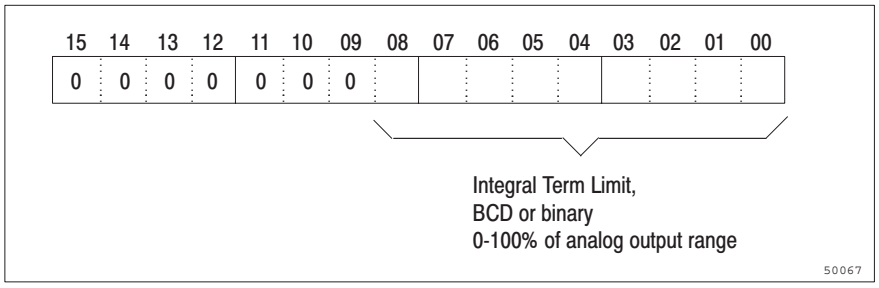
ATTENTION: To guard against equipment damage, we recommend that you exercise extreme care when operating an axis with PID error checking disabled.

Integral Term Limit (words 16 and 45)

The integral term limit parameter determines the maximum value that the integral term of the PID algorithm can obtain. You use this parameter for alarms and/or limiting.

The integral term limit prevents the integral term from causing maximum analog output if there is an undetected analog or hydraulic fault (e.g., if the hydraulic pump is off).

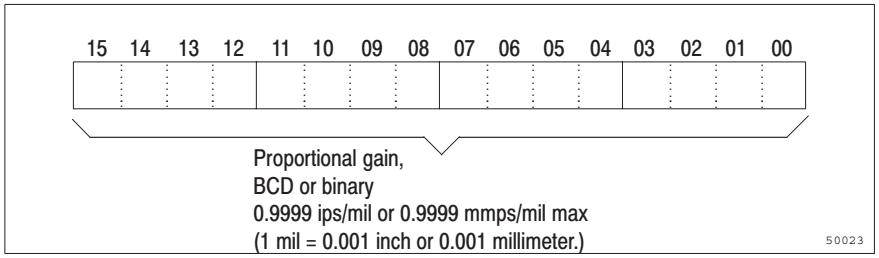
Figure 7.22
Integral Term Limit Word



Proportional Gain (words 17 and 46)

The module uses the proportional gain factor K_P at axis speeds below the gain break speed.

Figure 7.23
Proportional Gain Word

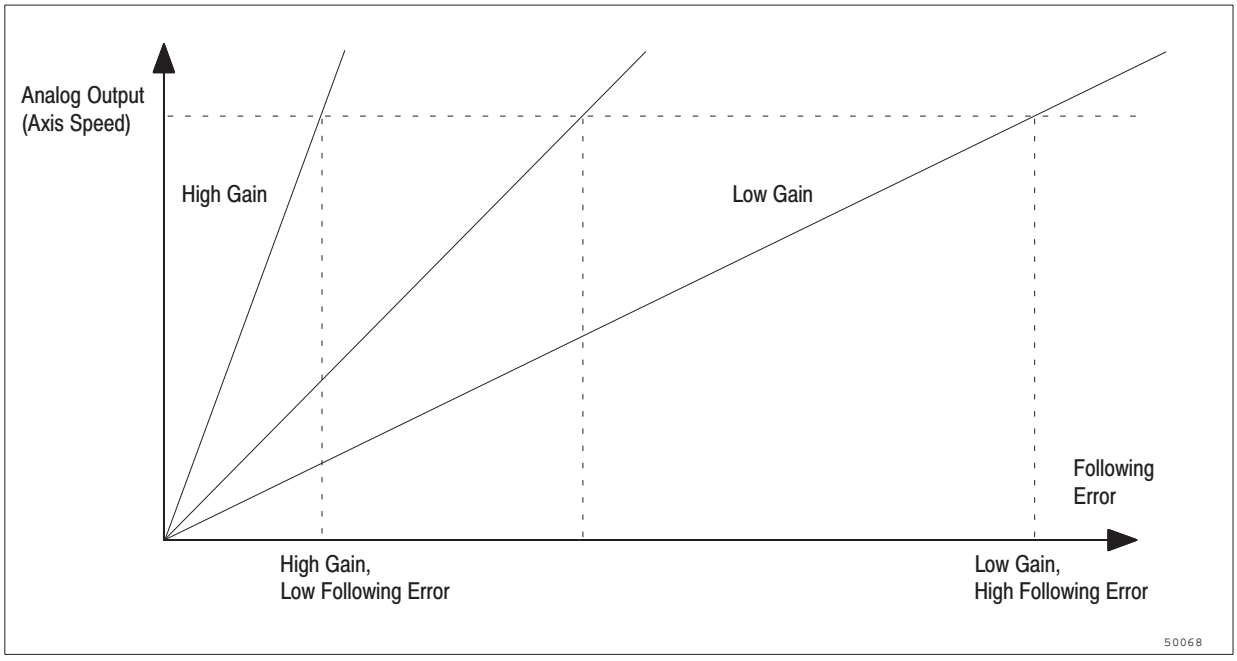

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Proportional gain,
BCD or binary
0.9999 ips/mil or 0.9999 mmps/mil max
(1 mil = 0.001 inch or 0.001 millimeter.)
50023

The proportional gain is defined as the ratio of the axis speed divided by the positioning error (or following error):

proportional gain = axis speed/positioning error

Proportional gain effects axis response to positioning commands. Figure 7.24 shows how different gain values affect system responsiveness.

Figure 7.24
Following Error vs Speed for Various Gains



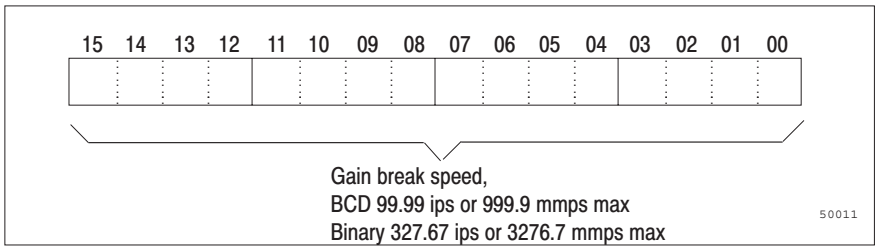
If gain is relatively high, following error will be relatively small, because the system will be more sensitive to changes in following error. If gain is low, following error becomes relatively larger, because the system is not as responsive to changes in following error. Choose a gain value to match the capabilities of your equipment and provide an adequate system response.

The proportional gain that you choose must provide a stable system and maintain desired positioning accuracy. If the gain is too high, the axis may overshoot programmed endpoints and oscillate around them. If the gain is too low, the axis may stop before it is within the desired in-position or PID bands.

Gain Break Speed (words 18 and 47)

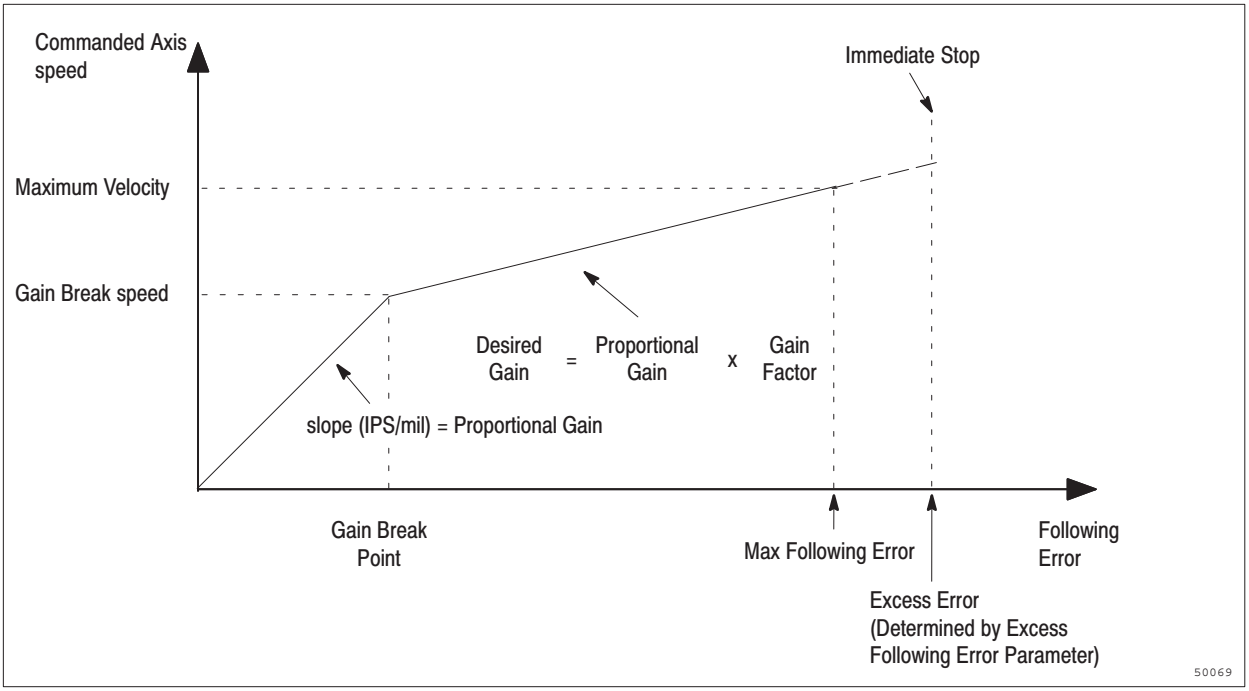
At speeds equal to and above the gain break speed, the proportional gain is increased or reduced by the gain factor parameter (words 19 and 48). Below the gain break speed, the proportional gain is unchanged.

Figure 7.25
Gain Break Speed Word



The gain break plot in Figure 7.26 illustrates the concept of gain break.

Figure 7.26
Gain Break Plot



Typically, at axis speeds below the gain break velocity, you would use a relatively high gain to allow precise axis positioning. By reducing the gain at axis speeds above the gain break speed, we can achieve better stability in some applications.

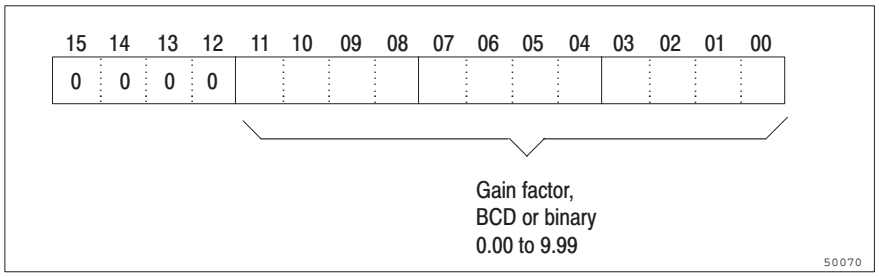
The gain break speed must not exceed the maximum velocities specified in the analog calibration constants.

If you don't want a gain break speed, set the gain break speed and gain factor parameters to zero. (If you set a non-zero gain factor and a zero gain break speed, the reduced or increased gain applies to all axis speeds.)

Gain Factor (words 19 and 48)

The gain factor parameter determines how much the proportional gain is reduced or increased at speeds above the gain break speed. It is expressed as a ratio of the new desired gain over the proportional gain.

Figure 7.27
Gain Factor Word



The gain factor must be less than 10.0. If you set it to zero, the proportional gain won't be reduced or increased at any axis speed.

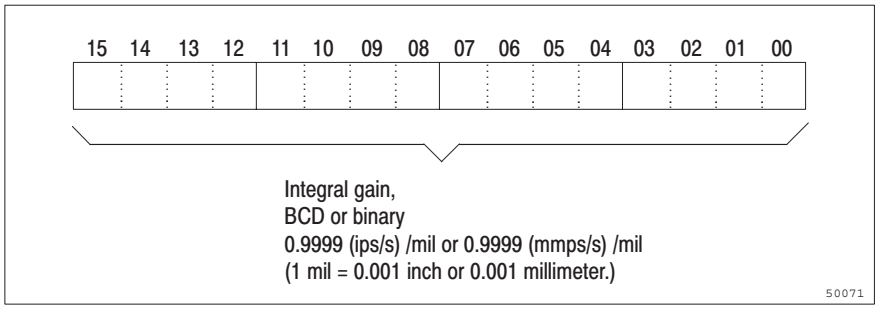
Example: To increase a proportional gain to 0.5 from 0.1 at speeds above the gain break speed:

gain factor = desired gain/proportional gain
= 0.5/0.1
= 5.00

Integral Gain (words 20 and 49)

The integral gain factor K_I is used by the integral component during final axis positioning, i.e., when the desired velocity is zero and the axis is in the PID band.

Figure 7.28
Integral Gain Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Integral gain,
BCD or binary
0.9999 (ips/s) /mil or 0.9999 (mmps/s) /mil
(1 mil = 0.001 inch or 0.001 millimeter.)
50071

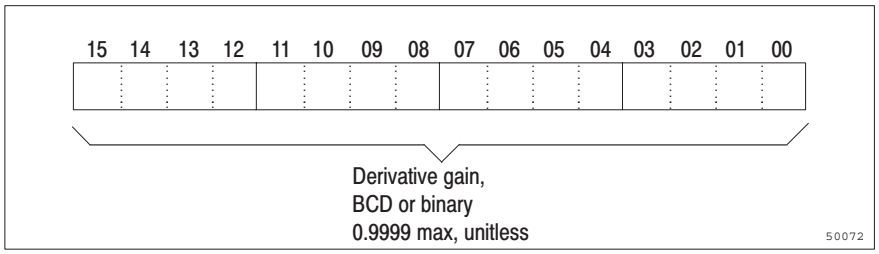
The module uses integral control to improve final positioning accuracy by making the system sensitive to the duration of positioning errors. If a positioning error exists, the integral term continues to alter the analog output until the axis overcomes inertia and reaches an accurate position.

The integral term alters response to positioning errors. If the integral gain is relatively high, the system will be more sensitive to positioning errors. However, if the gain is too high, the axis may overshoot and oscillate around programmed endpoints. On the other hand, if the gain is too low, the system will take longer to compensate for positioning errors.

Derivative Gain (words 21 and 50)

The derivative gain factor K_D is used by the derivative component during final axis positioning, i.e., when the desired velocity is zero and the axis is in the PID band.

Figure 7.29
Derivative Gain Word



The module uses derivative control to increase system stability by opposing changes to positioning error. Derivative gain reduces overshoots and oscillations around an endpoint caused by proportional and integral control, but too large a derivative gain will actually cause oscillations instead of reducing them.

Derivative control is also very susceptible to electrical noise. We recommend that you set derivative gain to zero if it isn't required to improve system performance.

Feedforward Gain (words 22 and 51)

The percentage of the axis velocity applied through feedforwarding determines the corresponding reduction in the following error. The magnitude of the feedforward contribution is calculated as follows:

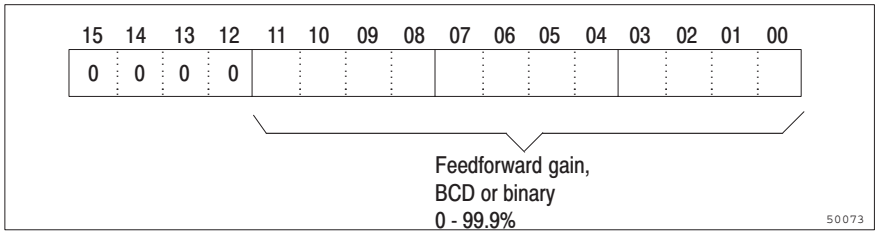
feedforward velocity = (Kf)(speed)

where,

Kf = feedforward gain

Speed = desired axis speed

Figure 7.30
Feedforward Gain Word



Without feedforwarding axis motion does not begin until the following error is large enough to overcome friction and inertia. The feedforward component generates a velocity command to move the cylinder almost immediately. This immediate response keeps the actual position closer to the desired position and thereby reduces the following error.

Example: Following Error Calculation

If axis movement is 5 ips and proportional gain (K_P) at that speed is 0.05 ips/mil, then

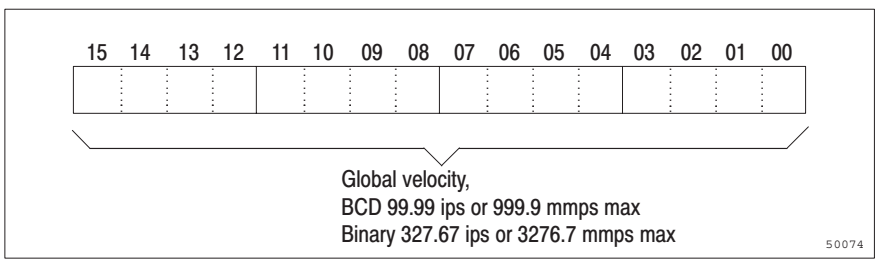
following error = (5)/(0.05) = 100 mil

A feedforward gain of 50% will reduce the following error to 50 mil (1 mil = 0.001 inches).

Global Velocity (words 23 and 52)

This parameter is used by setpoints and motion segments that use the global velocity. The global velocity must not exceed the maximum specified by the greater of either analog calibration constant.

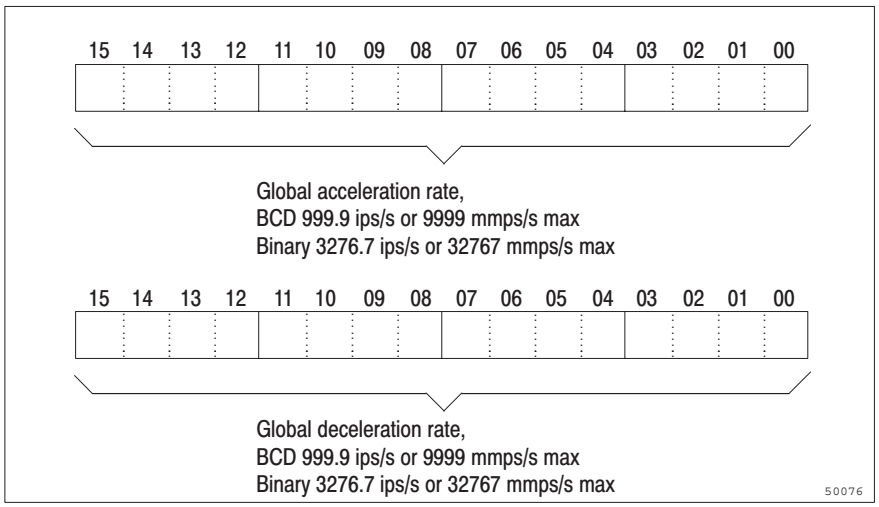
Figure 7.31
Global Velocity Word



Global Acceleration/Deceleration (words 24, 25 and 53, 54)

This parameter specifies the acceleration and deceleration rate the module uses for all jogs and for those setpoint and motion segment moves that do not use local acceleration and deceleration rates. The deceleration value is also used for executing slide stops during manual mode.

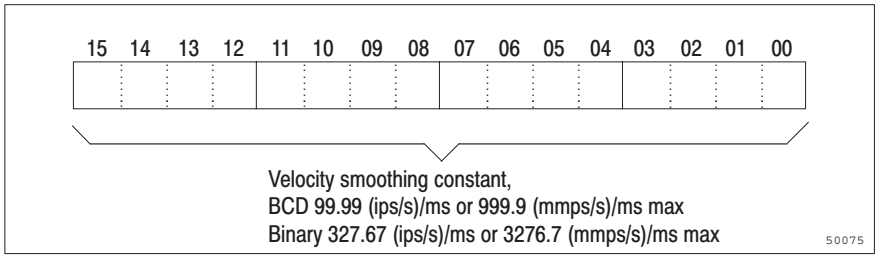
Figure 7.32
Global Acceleration/Deceleration Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Global acceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Global deceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50076

Velocity Smoothing (Jerk) Constant (words 26 and 55)

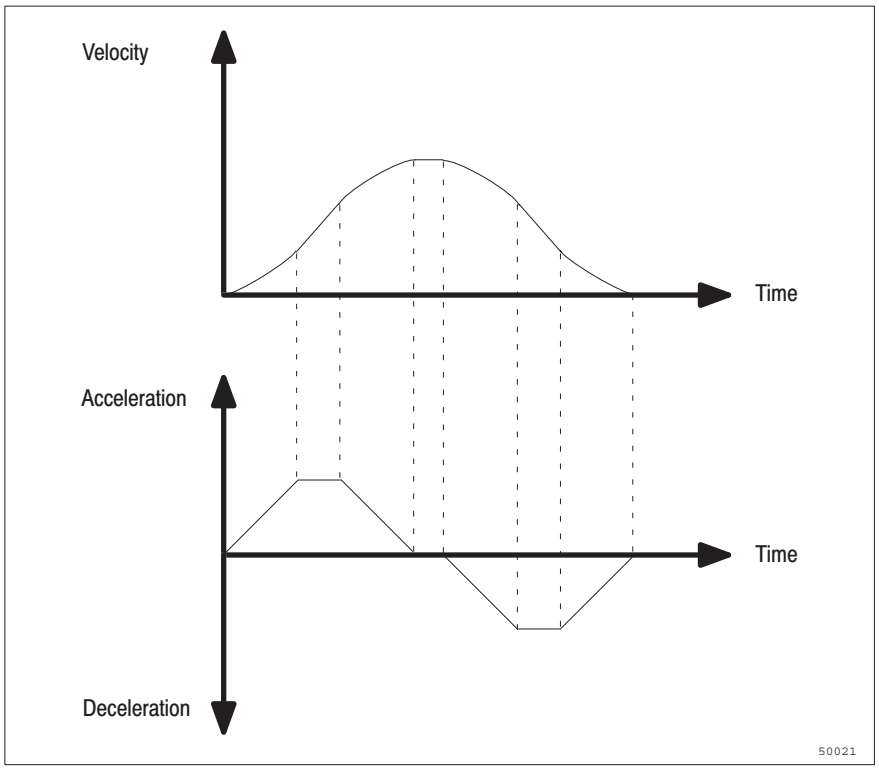
The velocity smoothing constant (also known as the jerk constant) specifies the maximum rate that the acceleration or deceleration can change. When the module commands a move, the acceleration and deceleration increases or decreases at the rate specified by this constant.

Figure 7.33
Velocity Smoothing (Jerk) Constant Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Velocity smoothing constant,
BCD 99.99 (ips/s)/ms or 999.9 (mmps/s)/ms max
Binary 327.67 (ips/s)/ms or 3276.7 (mmps/s)/ms max
50075

The velocity smoothing constant determines how quickly the system will change its acceleration and deceleration. The higher the value, the more quickly acceleration and deceleration will change. A higher, faster value produces jerkier motion, while a lower, slower value produces a smoother transition. The following diagrams demonstrate the effect of the velocity smoothing constant.

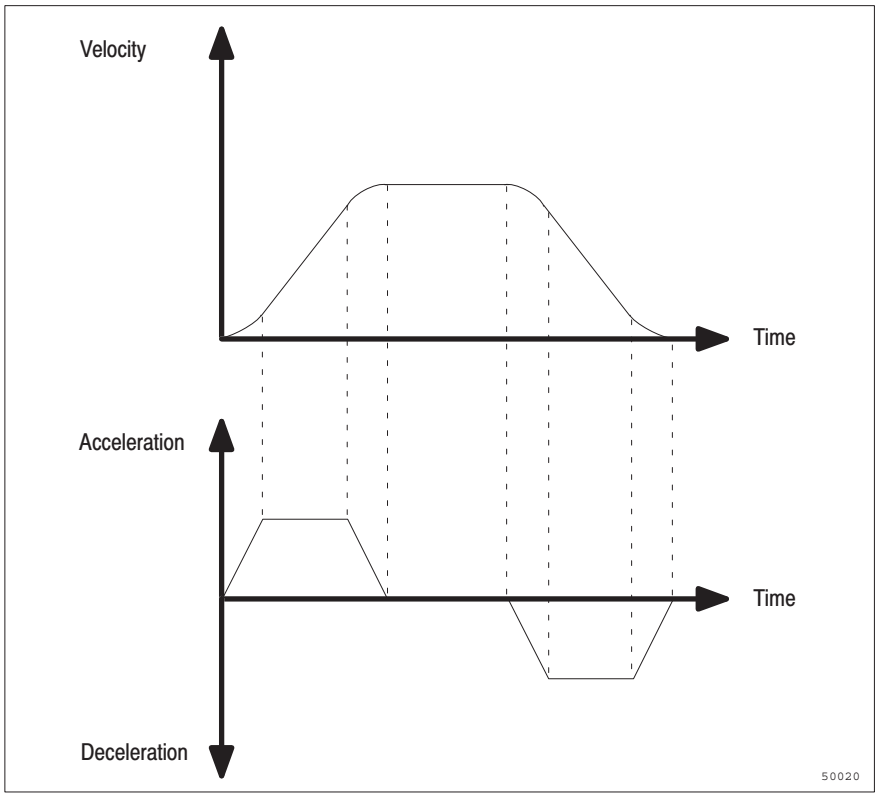
Figure 7.34
Lower Velocity Smoothing Constant



If the velocity smoothing constant is set to zero, velocity smoothing is disabled, and acceleration and deceleration will change instantaneously. The zero setting produces the jerkiest motion.

Because of the smoother motion that can be achieved by using this feature, you can achieve higher acceleration and deceleration levels without jerk. The smoother motion also results in less wear and tear and less tendency to overshoot the intended end position.

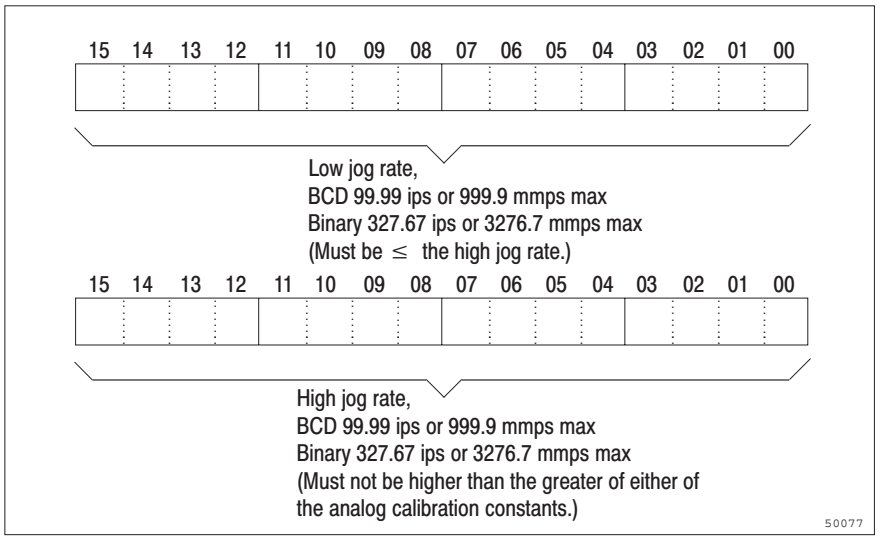
Figure 7.35
Higher Velocity Smoothing Constant



Jog Rate (Low and High) (words 27, 28 and 56, 57)

The jog rate words define the low and high speeds for software and hardware initiated jogs in either direction. By setting the jog rate select bit in the command block, you select high or low jog rate.

Figure 7.36
Jog Rate (Low and High) Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Low jog rate,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
(Must be ≤ the high jog rate.)
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
High jog rate,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
(Must not be higher than the greater of either of
the analog calibration constants.)
50077

Important: The low jog rate must be lower than or equal to the high jog rate. Both rates must be lower than the greatest of either maximum velocity, as specified by the analog calibration constants.

Reserved (words 29, 30 and 58, 59)

Words 29, 30, 58 and 59 are reserved for future use. You must set them to zero.

Setpoint Block

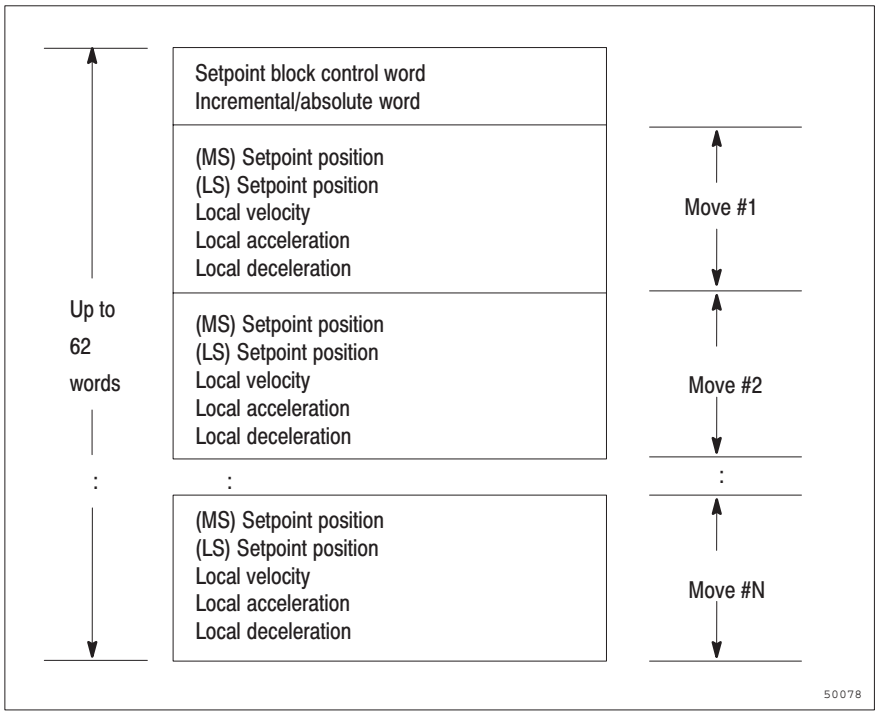
You use the setpoint block to define multiple setpoint moves. If you only need one setpoint for an axis, you can define it in the command block and not use the setpoint block at all.

You would normally define one setpoint block for each axis, but if your application requires that both axes use the same setpoint moves–i.e., both axes move in parallel–you can specify both axes in the setpoint block control word and use a single setpoint block.

Each setpoint block uses two words of overhead plus five words for each setpoint. Because the setpoint block can define from 1 to 12 setpoint moves for each axis, a setpoint block can be from 7 to 62 words long.

Each setpoint move defines the target axis position, velocity, acceleration, and deceleration. Figure 7.37 shows setpoint block word assignments.

Figure 7.37
Setpoint Block Word Assignments



The module internally maintains information on each of the 12 setpoints controlled by the setpoint block. On powerup or after a reset command each of these internal setpoints is disabled. The programmable controller must redefine one or all of these setpoints through the setpoint block. Therefore, the size of the setpoint block depends on the number of setpoints to be modified.

The first word in the setpoint block is the setpoint block control word. This control word identifies the block as a setpoint block, specifies that it applies to axis 1 or 2, and gives the number of setpoint moves contained in the block (up to 12).

Once the setpoint block is successfully processed by the module, you can command movement to selected setpoints in auto mode. You do this by specifying the setpoint number (1 to 12) in the command block.

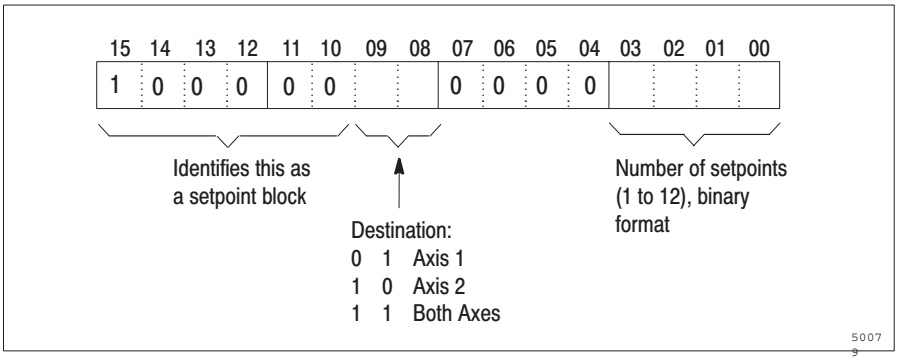
Moves to any setpoint are only possible in auto mode. If you select manual mode, only jog moves are possible.

The setpoint position is in metric or inches, depending on the option which you selected in the parameter block. The maximum range for programmable setpoint position is -799.900 to +799.900 inches (-7999.00 to +7999.00 millimeters).

Setpoint Block Control Word (word 1)

The setpoint block control word identifies the block as a setpoint block, specifies the axis or axes for which the setpoints are intended, and indicates the number of setpoints defined in the block.

Figure 7.38
Setpoint Block Control Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
1 0 0 0 0 0 0 0 0 0 0
Identifies this as a setpoint block
Destination:
0 1 Axis 1
1 0 Axis 2
1 1 Both Axes
Number of setpoints (1 to 12), binary format
50079

Bit 8 – Axis 1

If this bit is set, the module applies setpoint moves defined in this block to axis 1.

Bit 9 – Axis 2

If this bit is set, the module applies setpoint moves defined in this block to axis 2.

Important: If bits 8 and 9 are both set, a single block transfer will update both axes. You should only set both bits if your application requires that both axes use the same setpoint moves, i.e., that both axes move in parallel.

Incremental/Absolute Word (word 2)

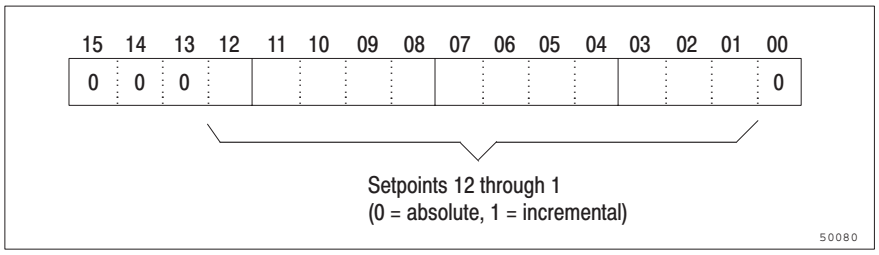
The incremental/absolute word defines the type of move performed by each setpoint in the setpoint block:

- **absolute** setpoint moves define a new endpoint in relation to the zero-position offset specified in the parameter block
- **incremental** setpoint moves define a new endpoint in relation to the current axis position (if the axis is stationary) or to the targeted endpoint (if the axis is still in motion)

Example: If the axis is stationary at +1 inch (from the zero-position offset), an absolute setpoint move with a position value of 2 inches will move the actuator 1 inch to the +2 inch position. An incremental setpoint move with a position value of 2 inches will move the actuator 2 inches to the +3 inch position.

Bits 1 through 12 control setpoints 1 to 12 with a one-to-one correspondence. If a bit is set to 1, the corresponding setpoint move is incremental. If it is reset to 0, the move is absolute. Bits 0, 13, 14, and 15 have no function and must be set to 0.

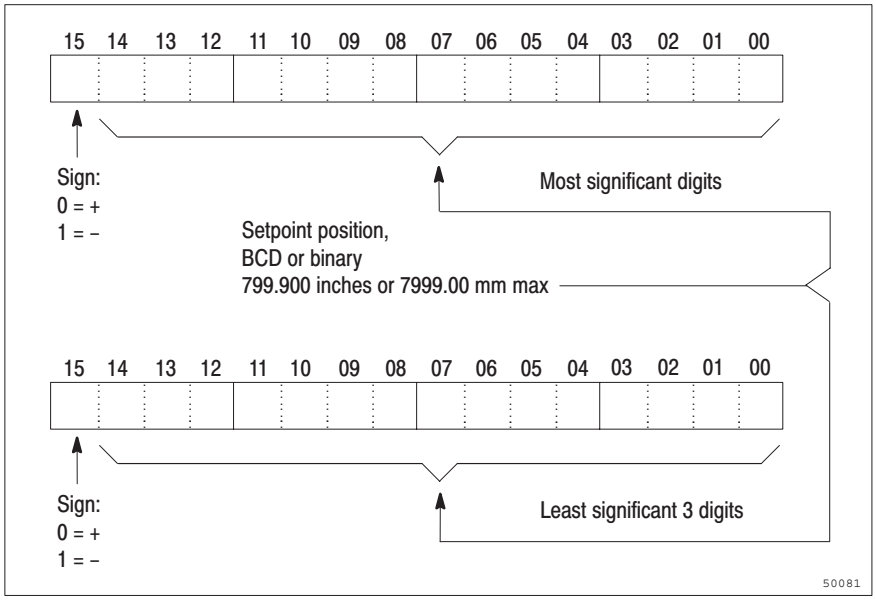
Figure 7.39
Incremental/Absolute Word



Setpoint Position

You must set the two setpoint position words for each setpoint move. Units are in either inches or metric and the format in BCD or binary as determined by the parameter block. The maximum programmable setpoint position is ±799.900 inches or ±7999.00 millimeters.

Figure 7.40
Setpoint Position Words

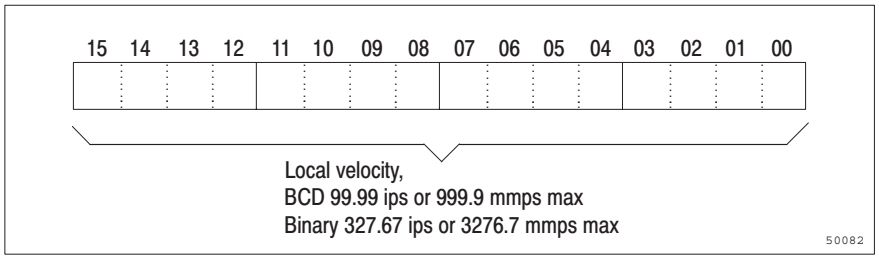

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Setpoint position,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50081

Important: If you select binary format, both words are represented as 2's-complement integers compatible with the PLC-5. See Appendix H for examples of these words.

Local Velocity

The local velocity word defines the velocity you want for the corresponding setpoint move. You can disable the local velocity by setting it to zero; the module will then use the global velocity.

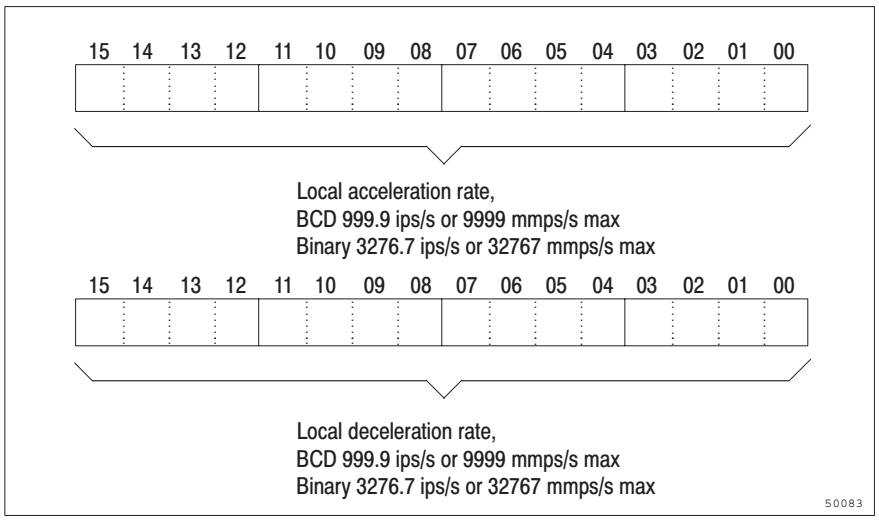
Figure 7.41
Local Velocity Words



Local Acceleration/Deceleration

The local acceleration and deceleration words define the acceleration and deceleration you want for the corresponding setpoint move. You can disable either or both of these parameters by setting them to zero. The module will then use the global parameter.

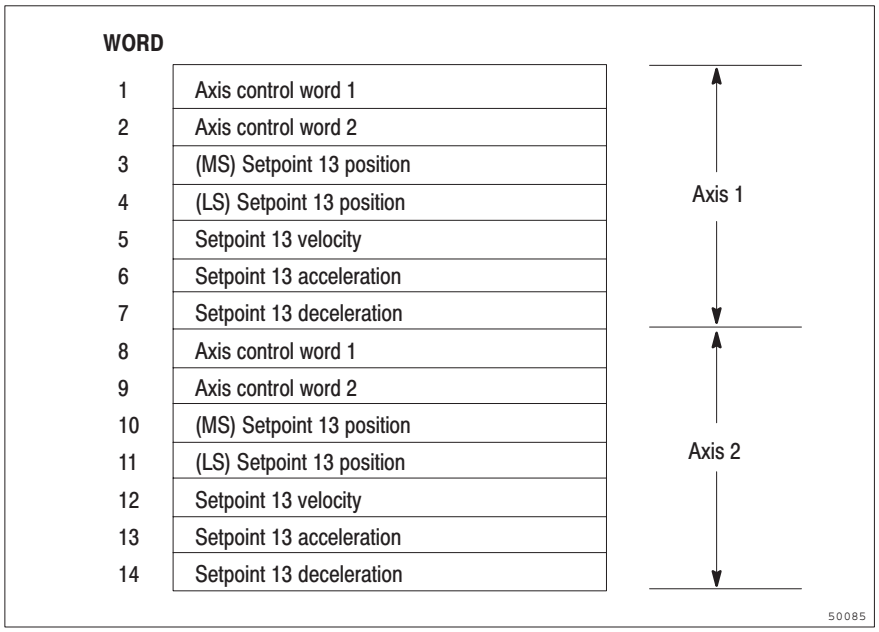
Figure 7.42
Local Acceleration/Deceleration Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local acceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local deceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50083

Command Block

You use the command block to control the module. It contains the words and bits necessary to initiate axis movement. Optionally, you can command movement to a 13th setpoint (in auto mode only). The command block contains two control words and five words defining setpoint 13 for each axis. If you are controlling one axis, the command block will be seven words long; if you are controlling two axes, the command block will be 14 words long.

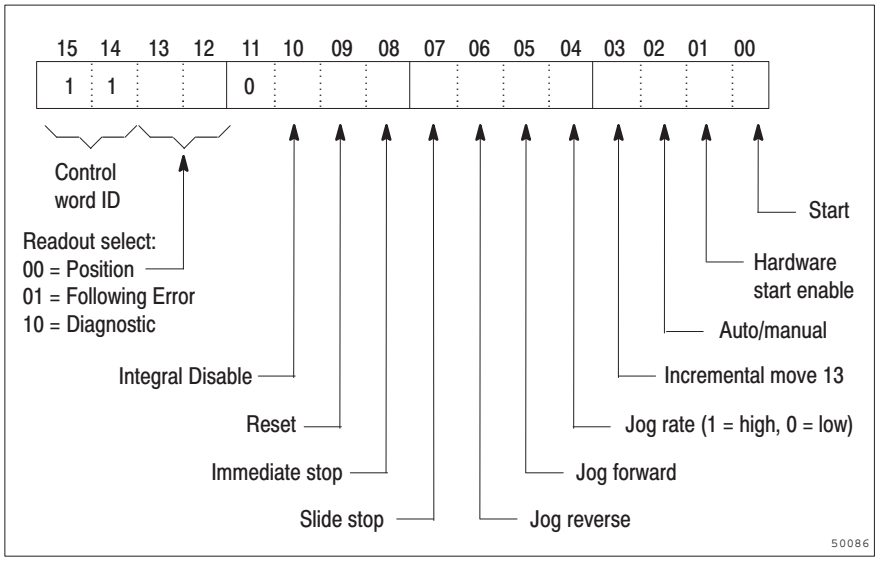
Figure 7.43
Command Block Word Assignments



Axis Control Word 1 (words 1 and 8)

The structure of axis control word 1 is shown below:

Figure 7.44
Axis Control Word 1



Bit 0 – Start

Bit 0 in the first axis control word is the start bit. The transition of this bit from low to high (0 to 1) signals a software start command. Upon receiving this command, the module begins the setpoint or motion segment move specified in axis control word 2. In manual mode, this transition causes the module to report a programming error in the status block and abort the move.

As long as the start bit remains high, (i.e. 1), you can also initiate software start commands and thus movement to a new position by:

- sending a new setpoint or motion segment number
- sending a new position, velocity, acceleration, or deceleration parameter for setpoint 13 when setpoint 13 is the selected setpoint

If you issue software start commands while the axis is in motion (often referred to as “changing setpoints on the fly”), the new setpoint or motion segment will supersede the old one. When the new endpoint can be achieved with the current axis motion, the axis will accelerate or decelerate to the final velocity for this new move. If the new endpoint is in the opposite direction of the current axis motion, the axis will stop before reversing and proceeding on to the new endpoint. The module applies the following criteria in using the old and new parameters when a software start command is issued with the axis in motion:

- the new position, velocity and acceleration parameters will always replace the old ones for the rest of the move.
- if velocity smoothing is disabled, the deceleration parameter can be increased but not decreased. Thus, if the new deceleration parameter is lower than the old one, it is ignored until axis motion stops.
- if velocity smoothing is enabled, the new deceleration parameter is ignored until axis motion stops.
- if the axis has to stop and change direction to achieve the new endpoint, the new deceleration parameter will always be used for the portion of the move in the opposite direction.
- for incremental setpoint moves, the module calculates the new endpoint relative to the endpoint for the old move.
- for incremental motion segment moves, the module calculates the new endpoint relative to the current axis position.
- if a programming error is detected in the new setpoint or motion segment data, the new move is aborted and motion continues to the endpoint specified in the old data.

Bit 1 – Hardware Start Enable

Bit 1 in the first axis control word is the hardware start enable bit. Setting this bit enables the hardware start input. If this bit is reset, the loop ignores hardware start commands. Note that the discrete inputs must also be enabled by the parameter block before the module will recognize hardware start inputs.

Hardware start commands aren't accepted until the loop is in auto mode. If the start/stop enhancement bit in the parameter block control word is reset, the hardware start commands are not accepted until the axis in in position. If this bit is set, the hardware start commands will be accepted during axis motion, just as software start commands are.

Bit 2 – Auto/Manual

The auto/manual bit selects the mode of operation for the axis:

- Off = manual mode
- On = auto mode

The module enters manual mode if either the hardware input or the auto/manual bit go low. If the hardware inputs are disabled (via the parameter block), the hardware auto/manual input is ignored.

Important: If the mode is changed while the axis is in motion, the status block indicates a programming error and the axis stops at the current deceleration rate. The programming error bit clears when the axis motion stops. If the mode change is still commanded after axis motion stops, the mode will change.

Bit 3 – Incremental Move 13

The incremental move 13 bit determines whether Setpoint 13 (in the command block) is an incremental or absolute setpoint move:

- Off = absolute
- On = incremental

See setpoint block, incremental/absolute word for an explanation of incremental and absolute moves.

Bit 4 – Jog Rate Select

The jog rate select bit determines the rate for jogs:

- Off = low jog rate

- On = high jog rate

If this bit changes state during a jog operation, the axis will accelerate or decelerate to the newly commanded rate at the global acceleration/deceleration rate programmed in the parameter block.

Bits 5 and 6 – Forward Jog and Reverse Jog

Turning on bit 5 causes axis motion in the forward direction. Similarly, turning on bit 6 causes axis motion in the reverse direction. In addition:

- the jog bits are valid only if the loop is in manual mode. They are ignored if it is in auto mode.
- the axis motion continues until you turn the jog bit off or until the actuator approaches the software travel limit (whichever occurs first).
- if the axis approaches the software travel limit and the jog bit is still on, axis motion will stop at the global deceleration rate.
- the axis speed is either the low jog rate or the high jog rate, as defined in the parameter block. You select the rate by the jog rate select bit (bit 4).
- the discrete jog inputs take priority over the corresponding jog bits.

You may switch jog directions while the axis is jogging. If, while the axis is jogging in the forward direction, you turn forward jog off and reverse jog on, the axis decelerates to zero and then jogs in the reverse direction.

Bit 7 – Slide Stop

Setting the slide stop bit causes the axis to decelerate to a stop at the programmed deceleration rate (local or global) for the current axis motion. After completion of a slide stop, the done bit in the status block turns on. The in-position bit also turns on when the axis enters the in-position band defined in parameter block. You can then initiate motion using jog commands, or by commanding setpoints or motion segments. The following rules apply:

- the module recognizes slide stop commands in either auto or manual mode
- slide stop commands have higher priority than setpoint or jog commands
- slide stop commands do not result in a loop fault

If the stop/start enhancement bit in the parameter block control word is set, then the axis will remain stationary while the slide stop bit is set. However, if the stop/start enhancement bit is cleared, the module will attempt to respond to new setpoint commands even though the slide stop bit is set.

Bit 8 – Immediate Stop

Setting the immediate stop bit causes the module to immediately set the analog output to zero and turn on OUTPUT 2 if it is configured as the loop fault output. To recover from an immediate stop condition, either issue a reset command or turn the I/O chassis power off and then back on. The following rules govern the immediate stop command:

- the module recognizes an immediate stop command in either auto or manual mode
- immediate stop commands have priority over slide stop, setpoint, or jog commands

Bit 9 – Reset

Setting the reset bit reinitializes the loop to powerup state. The module recognizes a reset command in either auto or manual mode. Reset commands have priority over setpoint or jog commands.

If the axis is in motion when this command is issued, the axis stops at the current deceleration rate. After the axis stops, the reset command is acknowledged through the status block (ready bit).

The reset command is similar to powerup. The module disables the axis until a new parameter block is received. However, unlike powerup, the other axis isn't affected.

If the discrete outputs, OUTPUT 1 and OUTPUT 2, are configured as programmable, resetting the module with the reset bit will not change the programmed configuration of those outputs. (See Chapter 9.)

Bits 10 – Integral Disable

To disable integral control, set bit 10 high. When integral control is disabled, the value of the integral term is maintained at the value it was prior to disabling, and integral windup does not occur. When integral control is re-enabled (set bit 10 low), the integral term behaves as it did prior to being disabled.

Bit 11 – Reserved

Bit 11 is reserved for future use. The PLC program must set it to zero.

Bits 12 and 13 – Readout Select

Bits 12 and 13 are the readout select bits. The third and fourth status words for an axis provide either current axis position, following error, or diagnostic information. The readout select bits determine which information is displayed in the status block. It is generally set to diagnostics information when using the extended status information.

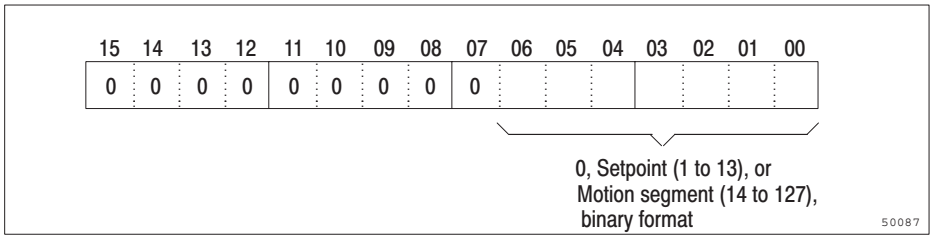
Bits 14 and 15 – Control Word ID

Bits 14 and 15 identify the control word of the command block.

Axis Control Word 2 (words 2 and 9)

The structure of the second axis control word is shown in Figure 7.45.

Figure 7.45
Axis Control Word 2



Bits 0 to 6 – Next Setpoint/Motion Segment

Bits 0 to 6 in the second axis control word contain the number of the next setpoint or motion segment you wish to initiate. See Chapter 9 for more information on motion segments.

- if all bits are 0 or if the axis is already at the commanded setpoint or motion segment, the axis remains unchanged.
- if the new setpoint/motion segment is in the opposite direction to the movement of the axis, the axis will stop at the deceleration rate for current motion before reversing and proceeding to the new setpoint.
- the module ignores the setpoint/motion segment until it recognizes a hardware or software start.
- setpoints 1 to 12 are defined in the setpoint block. If you specify setpoint 13, you must define it in the command block.

Bits 7 to 15 – Reserved

Bits 7 to 15 are reserved for future use. The programmable controller program must set them to zero.

Setpoint 13 Words (words 3 to 7 and 10 to 14)

The setpoint 13 words control position, velocity, acceleration, and deceleration for setpoint 13. These words have the same format as those for the other setpoints in the setpoint block.

Chapter 8

Initializing and Tuning the Axes

Before you load an application ladder logic program into the programmable controller, you should follow the procedures in this chapter to initialize and tune the movement of each axis. A simple ladder logic program, QB_SETUP, provided on the accompanying disk, is intended to simplify the initial integration process. Note that all specific references to data tables in the following procedures refer to QB_SETUP. If you prefer, you can write your own program for this process. In any case, initializing and tuning should be completed for each axis before bringing the module online. The steps are as follows:

- adjust the null on each servo valve
- load the ladder logic program and initialize the parameter block
- verify the analog output polarity
- tune the loop parameters for the axis
- copy the new loop parameters for the axis into the data table of the application program

Important: For these procedures to work, servo valves and motors must be capable of controlling axis motion according to your requirements. The module cannot overcome inherent limitations of the proportional valve or axis mechanisms.

Before You Begin

Before you perform these procedures, you must:

- disconnect the actuator from the tool or workpiece. If you can't disconnect the actuator, make sure that sudden or erratic motion will not result in equipment damage or personnel injury before continuing.
- if the system uses axis overtravel limit switches, verify their operation.
- supply power for and wire up the transducer, analog outputs and the I/O chassis containing the programmable controller and the module. The discrete inputs and outputs do not have to be powered up or wired in.

Adjusting the Servo Valve Nulls

The first step in initializing the module is to adjust the null on each servo valve. To do so, carry out the following steps:

1. Turn off axis power.
2. Disconnect the servo valve from the module.
3. Start the hydraulic pump and check the system pressure.

ATTENTION: To guard against possible injury keep all personnel clear of the axis and have a competent person standing by to disconnect axis power and stop the flow of hydraulic fluid, if necessary.

---

4. Adjust the servo null adjustment screw until there is no detectable movement in the cylinder. In some applications it is important that the axis should drift towards one end of the stroke if an electrical or servo fault occurs. In these cases, the valve should be nulled so that the axis drifts at one quarter inch per second (0.25 ips) in the desired direction. Make sure the blocking valve or safety plate is energized.
5. Turn off the hydraulic pump.

ATTENTION: Keep the axis near its center of travel. Running the axis into its mechanical stops could damage equipment.

---

6. Re-connect the servo valve.

Initializing the Parameter Block

If you load the ladder logic program QB_SETUP (on the accompanying disk) into the programmable controller to initialize the parameter block, the default settings will be as shown in Table 8.A. These settings should work for most applications. Before you load the program, check that the application dependent parameters noted in Table 8.A are correct for your system. If they are not, edit the data tables to correct the parameter values. It is strongly recommended that you also use the axis tuning procedures that follow, to ensure that the settings you have selected are correct for your application. QB_SETUP assumes that the module is mounted in slot 0 of rack 0. If your module is mounted differently, edit the ladder logic program to reflect this. The ladder logic and data tables for QB_SETUP are illustrated in Figure 8.1 to Figure 8.3.

Important: If you change any parameter block data, you must either cycle power to the I/O chassis or send a reset command to the module by toggling bit 9 at data table address N45:131.

Table 8.A
Default Parameter Block Settings

Parameter	Suggested Values		Comments
	Inches	**Metric**	
Analog range	100	100	Check servo valve rating
+Analog calibration constant	2500	6350	Set to maximum velocity in positive direction
–Analog calibration constant	2500	6350	Set to maximum velocity in negative direction
(MS) Transducer calibration constant	9	3	Multiply this by the number of circulations
(LS) Transducer calibration constant	500	5600	Multiply this by the number of circulations
(MS) Zero-position offset	-5	-12	Application dependent
(LS) Zero-position offset	0	-700	Application dependent
+Software travel limit	210	533	Set in front of mechanical stop in (+) direction
–Software travel limit	-10	-25	Set in front of mechanical stop in (–) direction
In-position band	100	254	Optional - adjust for your application
PID band	500	1270	Rarely necessary to use a smaller value
Deadband	0	0	Rarely necessary or desirable
Excess following error	1000	2540	Non-zero value important for safety
Maximum PID error	1000	2540	Non-zero value important for safety
Integral term limit	10	10	Important to prevent integral windup
Proportional gain	600	600	Follow tuning procedure
Gain break speed	0	0	Rarely necessary
Gain factor	0	0	Rarely necessary
Integral gain	420	420	Set to 70% of proportional gain value
Derivative gain	420	420	Set to 70% of proportional gain value
Feedforward gain	300	300	Use these values or follow optional tuning procedure
Global velocity	1000	2540	Must be less than maximum velocity
Global acceleration	1000	2540	Application dependent
Global deceleration	1000	2540	Application dependent
Velocity smoothing constant	300	762	Application dependent
Low jog rate	100	254	Must not exceed high jog rate
High jog rate	1000	2540	Must be less than maximum velocity
Reserved	0	0	
Reserved	0	0	

Make sure that you have set the analog configuration switches correctly and that you have entered the correct range into the parameter block.

ATTENTION: Incorrect analog output configuration can cause sudden high-speed motion. Incorrect analog output polarity will cause the axis to accelerate out of position when you close the loop.

Figure 8.1
Parameter Block Data Table

Project Name: QB_SETUP - Axis 1 Page____ of ____
Designer: _____ Address ____ of ____
Date:_____ Axis No. 1 Block Description: Parameter

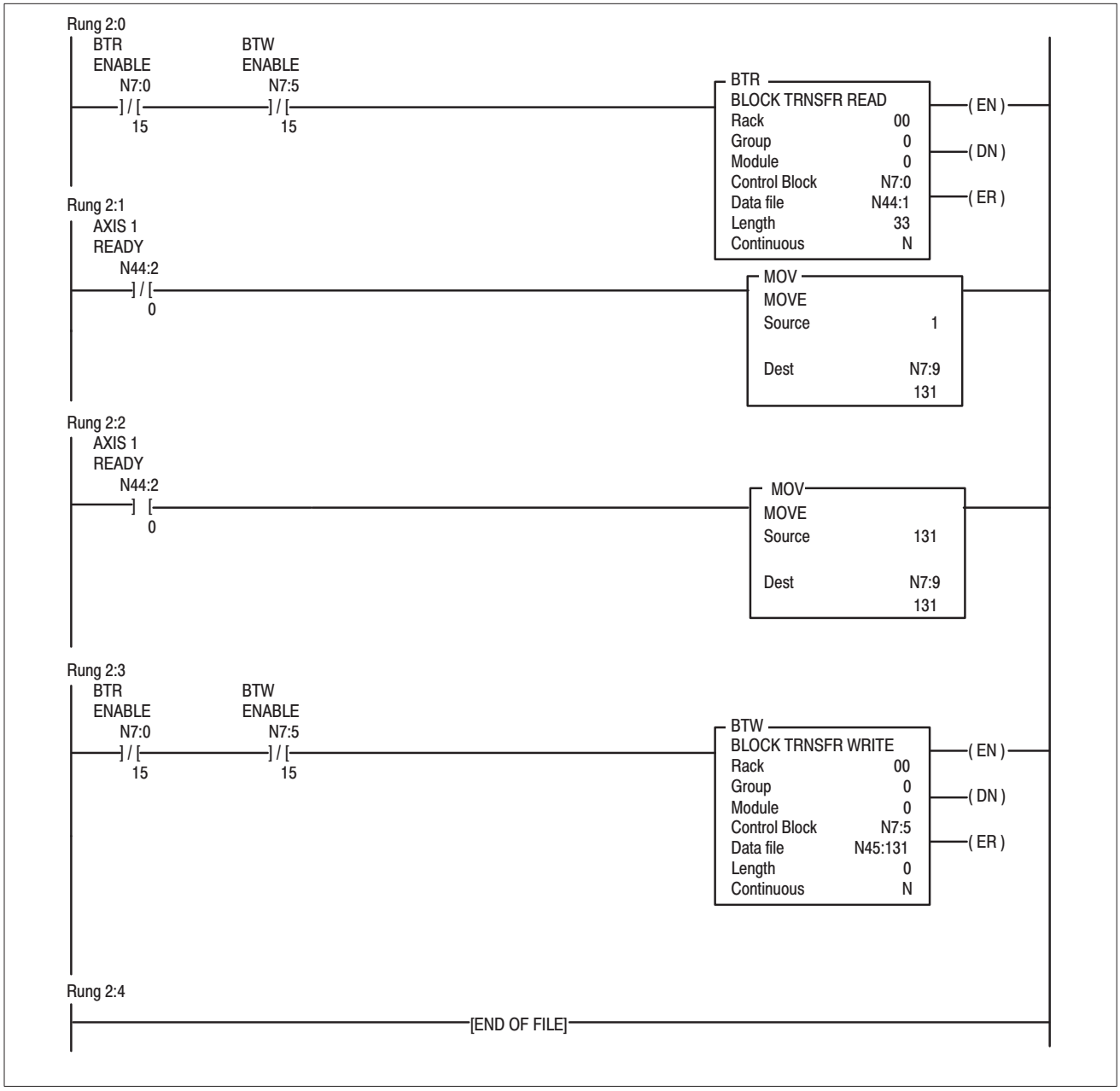
Data Table Address	Position	File Data				Description
N45:1	**1**	4	1	0	9	Parameter control word (Axis 1, inches, BCD)
2	**2**	0	1	0	0	Analog range (100%)
3	**3**	2	5	0	0	+Analog calibration constant. (25.00 ips)
4	**4**	2	5	0	0	–Analog calibration constant (25.00 ips)
5	**5**	0	0	0	9	(MS) Transducer calibration constant (9.0500 microsec/inch)
6	**6**	0	5	0	0	(LS) Transducer calibration constant
7	**7**	8	0	0	5	(MS) Zero-position offset (-5.000 inches)
8	**8**	0	0	0	0	(LS) Zero-position offset
9	**9**	0	2	1	0	+Software travel limit (21.0 inches)
10	**10**	8	0	1	0	–Software travel limit (-1.0 inches)
11	**11**	0	1	0	0	In-position band (0.100 inches)
12	**12**	0	5	0	0	PID band (0.500 inches)
13	**13**	0	0	0	0	Deadband (0.000 inches)
14	**14**	1	0	0	0	Excess following error (1.000 inches)
15	**15**	1	0	0	0	Maximum PID error (1.000 inches)
16	**16**	0	0	1	0	Integral term limit (10%)
17	**17**	0	6	0	0	Proportional gain (0.0600 ips/mil)
18	**18**	0	0	0	0	Gain break speed (0.00 ips)
19	**19**	0	0	0	0	Gain factor (0.00)
20	**20**	0	4	2	0	Integral gain (0.0420 (ips/s)/mil)
21	**21**	0	4	2	0	Derivative gain (0.0420)
22	**22**	0	3	0	0	Feedforward gain (30.0%)
23	**23**	1	0	0	0	Global velocity (10.00 ips)
24	**24**	1	0	0	0	Global acceleration (100.0 ips/s)
25	**25**	1	0	0	0	Global deceleration (100.0 ips/s)
26	**26**	0	3	0	0	Velocity smoothing constant (3.00 (ips/s)/ms)
27	**27**	0	1	0	0	Low jog rate (1.00 ips)
28	**28**	1	0	0	0	High jog rate (10.00 ips)
29	**29**	0	0	0	0	Reserved
30	**30**	0	0	0	0	Reserved

Figure 8.2
Command Block Data Table

Project Name: QB_SETUP - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Command

Data Table Address	Position	File Data				Description
N45: 131	**1**	E	0	0	4	Axis control word 1 (Diagnostic, auto)
132	**2**	0	0	0	0	Axis control word 2
133	**3**	0	0	0	0	(MS) Setpoint 13 position
134	**4**	0	0	0	0	(LS) Setpoint 13 position
135	**5**	0	0	0	0	Setpoint 13 velocity
136	**6**	0	0	0	0	Setpoint 13 acceleration
137	**7**	0	0	0	0	Setpoint 13 deceleration

Figure 8.3
Program Rungs for QB_SETUP



Verifying Analog Output Polarity

You should verify the analog output polarity using low speed open-loop jogs as follows:

ATTENTION: Incorrect analog output polarity will cause the axis to accelerate out of position when then loop is closed. The analog output polarity is affected by both the wiring on terminals 29/31 and the sign of the analog range in the parameter block.

---

1. Turn off axis power.
2. Disconnect the transducer gate leads from the wiring arm terminals 1/3.
3. Start the hydraulic pump.
4. Apply power to the I/O chassis, the analog output, the transducer interface and the discrete inputs. Don't power the axis that you are not integrating.
5. Clear bit 2 at N45:131 to place the axis in manual mode. Use the forward and reverse jog bits in axis control word 1 of the command block (bits 5 and 6 at data table address N45:131) to command jogs in the positive and negative directions. Check that a forward jog moves the axis in the positive direction as defined by the zero-position offset, and a reverse jog moves the axis in the negative direction.
6. If the direction of motion is incorrect, either reverse the analog connections at the module wiring arm or change the sign of the analog range word in the parameter block. Recheck that the direction of motion is correct.
7. Turn off axis power and the hydraulic pump.
8. Re-connect the transducer gate leads to the wiring arm terminals.

Verifying Transducer Calibration Constants

The module uses the transducer calibration constant to convert pulses received from the transducer to a position in real world units. The transducer constant is usually stamped on the side of the transducer. However, sometimes the stamp is not easily accessible or you may not be confident that it is accurate. You can use the following procedure to determine the transducer calibration constants for the transducers.

You will need a six inch or longer caliper to measure axis position. The length and accuracy of the caliper determines the accuracy of the transducer calibration constant. Longer or more accurate instruments will produce better results.

1. Use Table 8.B to determine your initial guess, based on the number of circulations programmed into the digital interface box.

Table 8.B
Transducer Calibration

Number of Circulations	Transducer Calibration Constant	
	Microsec/Inch	Microsec/mm
1	9.0500	0.35600
2	18.1000	0.71200
3	27.1500	1.06800
4	36.2000	1.42400
5	45.2500	1.78000
6	54.3000	2.13600
7	63.3500	2.49200
8	72.4000	2.84800

2. Enter the transducer calibration constant from the table into the parameter block for the axis. Send the new parameter block to the module.

3. With the hydraulic pump off, mark the position of the axis relative to a stationary structure. You will need to measure the change in the axis position to within 0.001 inches.

4. Record the *initial axis position* value from the module. This value is in the status block words 12 and 13 at N44:12 and N44:13 for axis 1.

5. Turn the hydraulic pump on and move the axis approximately six inches, using one of the following three techniques:

- if you have previously tuned your axis use the command block (at N45:131) to either jog the axis or perform a setpoint 13 move
- if you have not previously tuned your axis, with axis power off, disconnect the servo valve from the module and use the servo valve nulling screw to move the axis
- if you cannot easily access the valve null screw, with axis power off, disconnect the transducer gate leads from the wiring arm. Restore power and then jog the axis at *one inch per second* using the jog bits. (This is called the open-loop jog.) With axis power off, reconnect the trandsucer gate leads. restore power and continue.

6. Turn the hydraulic pump off.

7. Using your caliper, measure the distance travelled by the workpiece, and record this value as the *actual distance*.

8. Record the *new axis position* value from the module. This value is in the status block words 12 and 13 at N44:12 and N44:13 for axis 1.

9. Subtract the *initial axis position* from the *new axis position* and record this as the *module distance*.

10. Divide the *module distance* by the *actual distance* and record this as *transducer calibration correction factor.*

11. Multiply the transducer calibration constant in the parameter block words 5 and 6 (N45:5 and N45:6) by the *transducer calibration correction factor* calculated in steps 9 and 10. Enter the result as the new transducer calibration constant and reset the axis by toggling bit 9 of the axis control word N45:131.

12. Repeat steps 3 through 11 to verify that the transducer calibration constant is correct.

13. Determine a valid set of parameters for the zero-position offset and the software travel limits and enter these into the parameter block.

Example: Verifying Transducer Calibration Constants

If the values for axis 1 are:

initial position	= 12.324
actual distance	= 6.423
new position	= 18.737
transducer calibration constant	= 9.0500

Then the module distance is:

18.737 – 12.324 = 6.413

The transducer calibration correction factor is:

6.413 / 6.423 = 0.99844
and the new transducer calibration constant is:

0.99844 x 9.0500 = 9.0359

Axis Tuning

Each axis needs to be tuned to allow for its specific mechanical and electrical characteristics. If you change system variables, such as hydraulic pressure, cylinder size or servo valve characteristics, you may need to re-tune your axis as well. Remember that every time you change parameters in the parameter block, you should reset the axis using bit 9 of the axis control word 1 (N45:131).

Analog Calibration Constants

It is important that the positive and negative analog calibration constants be optimized first. These are critical constants, since the PID and feedforward gains use them to correct for directional differences in system performance.

Follow these steps to adjust the analog calibration constants:

1. Start the hydraulic pump and check the system pressure. Apply power to the I/O chassis and the axis.

ATTENTION: To guard against possible injury: Keep all personnel clear of the axis. Have a competent person standing by to disconnect axis power and stop the flow of hydraulic fluid if necessary.

---

2. If you have previously tuned the axis, you can leave the PID and feedforward gains unchanged. Otherwise initialize the loop parameters as follows:

Proportional gain:	Kp = 0.0100 ips/mil
Integral gain:	K_I = 0.0100 (ips/s)/mil
Derivative gain:	K_D = 0
Feedforward gain	K_F = 0
Gain break speed	= 0 ips
Gain factor	= 0
Integral term limit	= 10%
PID band	= 0.500
Deadband	= 0
Maximum PID error	= 1.000 in
Excess following error	= 1.000 in

3. Initialize both analog calibration constants to the maximum velocity possible for the servo cylinder selection. For hydraulic systems with cylinders with circular cross sections, the maximum velocity can be approximated using the following formula:

$$\text{Velocity} = 4.9\ Q/B^2 \text{ (inches per second)}$$

where: Q = flow rate in gallons per minute
B = bore of cylinder in inches

Example: Maximum Velocity Calculation

If you have a cylinder with a 2 inch bore (inside diameter) and a servo valve rated for 10 gallons per minute, the maximum velocity is approximated as follows:

Velocity = $4.9 \times 10/2^2 = 12.25$ ips

4. Use bits 5 and 6 of axis control word 1 (N45:131) to jog the axis back and forth between the software travel limits until the maximum velocity stabilizes in the status block.

Important: The maximum velocity predictions will vary slightly for moves at different velocities due to non-linearities in the hydraulic system. If it is critical that the module perform best at a particular velocity, that velocity should be used to determine the optional analog calibration constants. Otherwise, it is recommended that you use a moderately low velocity (10% of the maximum velocity) for this purpose. Adjust this velocity in N45:27.

5. The maximum positive velocity (words N44:30 in the status block) is the positive analog calibration constant. The maximum negative velocity (words N44:32 in the status block) is the negative analog constant. Enter these values into the parameter block (words N45:3, and N45:4).

Important: If the maximum velocity returned by the module is dramatically different than your initial guess, check the analog output switches and the analog range word.

Feedforward Gain

The feedforward gain can dramatically reduce the following error during a move. The effectiveness of the feedforward gain is dependent on the accuracy of the analog calibration constants, the linearity of the servo valve, and system responsiveness.

You can usually achieve acceptable results by selecting a conservative feedforward gain of 30%. For more precise velocity control, adjust the feedforward gain as follows:

1. Start the hydraulic pump and check the pressure.

2. Initialize the loop gains as follows:

Proportional gain:	K_P = 0.0050 ips/mil
Integral gain:	K_I = 0
Derivative gain:	K_D = 0
Feedforward gain	K_F = 20.0%

3. Initiate a move using setpoint or jog commands. Increase the feedforward gain until the axis begins to overshoot, i.e., to exceed the desired end position.

4. Decrease the feedforward gain by 10%. This reduced value will result in a more stable performance of the axis.

PID Loop Gains

The next step in tuning the system is to find the appropriate PID gains. The following procedure requires that you increase the proportional gain until the axis oscillates. Do not use this procedure if your equipment cannot tolerate oscillation.

First, determine the proportional gain:

1. Initialize the loop parameters to the following defaults:

K_P	= 0.0200 ips/ml
K_I	= 0
K_D	= 0

The gain break speed and gain factor should be zero.

2. Initiate a jog or setpoint move with the highest acceleration and deceleration that you plan to use.

3. Increase the proportional gain using increments of 0.0100 and repeat step 2 until continuous oscillations result.

4. Decrease the proportional gain by 30%. This will result in stable performance of the axis. In most cases, a proportional gain between 0.0300 and 0.1500 ips/mil will give optimal performance.

$K_{P(unstable)}$	= lowest proportional gain at which continuous oscillations occur
KP	= 0.7 x $K_{P(unstable)}$

Next determine the integral and derivative gains. The following procedure will produce near-optimal results for most systems.

5. Set the integral gain equal to 70% of the proportional gain at which continuous oscillations occurred (see step 3).

$K_I = 0.7 \times K_P$

6. Set the derivative gain equal to 70% of the proportional gain at which continuous oscillations occurred (see step 3). This small derivative gain is recommended to improve axis stability.

$K_D = 0.7 \times K_P$

7. Check the axis stability under all load conditions. In some instances, you may have to adjust the proportional, integral and derivative gains slightly or use the gain reduction factor to obtain the desired results.

Example: Optimal PID Loop Gain Calculation

If continuous oscillations occur with a proportional gain of 0.1000 ips/mil and the integral and derivative gains set to zero, the optimal gains are calculated as follows:

$K_{P(unstable)}$ = 0.1000 ips/mil
K_P = 0.7 x 0.1000 = 0.0700 ips/ mil
K_I = 0.7 x 0.0700 = 0.0490 (ips/s)/mil
K_D = 0.7 x 0.0700 = 0.0490 unitless

Update the Application Program

As a final step, you should copy the parameter block data determined above, into the data tables for your application. The tuning and integration procedure used axis 1, but you can perform a similar procedure with axis 2.

Chapter 9

Advanced Features

The advanced features of the Linear Positioning Module enable you to create complex movement profiles, synchronize multiple axes, and perform cam-emulation. They are not required in order to use the module, and should only be used once you fully understand how to initiate and control motion using setpoints.

Motion Block

You can implement complex movement profiles using the motion segments of the motion block. Such profiles can be executed without the intervention of the programmable controller. The axis moves from one segment to another using a user-selectable trigger for each motion segment. You define a movement profile by downloading one or more motion blocks from the programmable controller to the module using block transfer writes. The user-controlled sequence of up to 114 motion segments is then started by specifying the starting motion segment in the command block.

Each motion segment defines a setpoint and a trigger condition. The setpoint defines the movement profile for a motion segment, while the trigger defines the conditions to be met to initiate execution of the next motion segment. The triggers include position, velocity and discrete inputs. Movements based on position triggers are called chained moves, while movements based on velocity triggers are called blended moves. A movement profile may contain blended moves, chained moves and moves based on discrete input triggers.

A motion block can contain a maximum of six motion segments. It begins with a motion block control word and may end with an optional programmable I/O control word. (See Figure 9.1.) The programmable I/O word lets you configure programmable input and output options.

ATTENTION: You must use the motion block with caution because, the programmable controller cannot necessarily predict the final axis position, since motion segments may be triggered without its knowledge. You should understand thoroughly the consequences of all the motion profiles which may occur after a motion block is initiated, and design the ladder logic to account for them.

Important: All segments in a motion block, and the programmable I/O word, become valid as soon as they are downloaded to the module. The one exception is a downloaded segment corresponding to the currently active motion segment. In that case, the active segment must complete its profile or meet its trigger conditions before the new segment becomes valid. You must ensure that the programmable controller uses the block transfer write toggle bit in the status block to accurately load the motion blocks because the module returns no status bits to indicate the current programmable I/O configuration or the currently loaded motion segments.

Figure 9.1
Motion Block Word Assignments

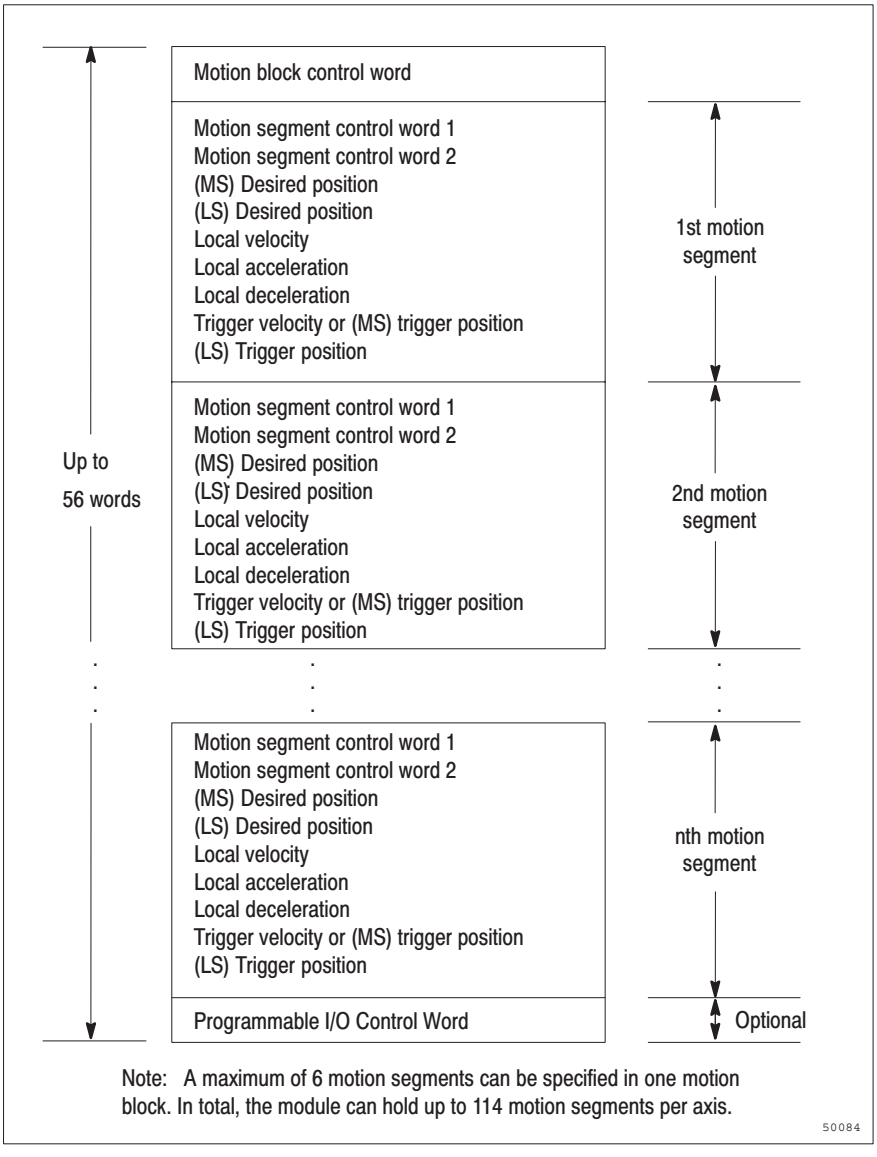
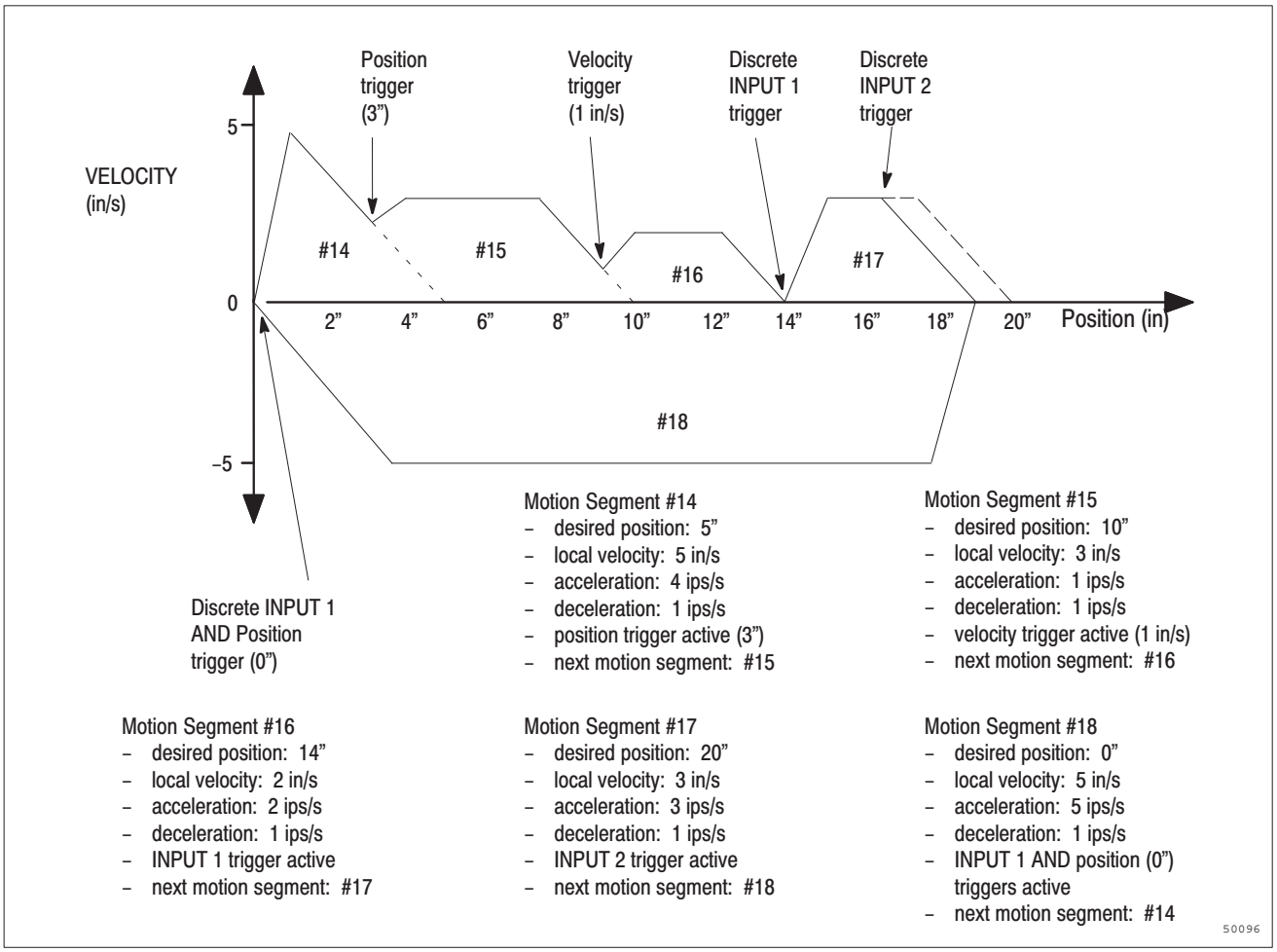

Motion block control word
Motion segment control word 1
Motion segment control word 2
(MS) Desired position
(LS) Desired position
Local velocity
Local acceleration
Local deceleration
Trigger velocity or (MS) trigger position
(LS) Trigger position
1st motion segment
Motion segment control word 1
Motion segment control word 2
(MS) Desired position
(LS) Desired position
Local velocity
Local acceleration
Local deceleration
Trigger velocity or (MS) trigger position
(LS) Trigger position
2nd motion segment
Up to 56 words
Motion segment control word 1
Motion segment control word 2
(MS) Desired position
(LS) Desired position
Local velocity
Local acceleration
Local deceleration
Trigger velocity or (MS) trigger position
(LS) Trigger position
nth motion segment
Programmable I/O Control Word
Optional
Note: A maximum of 6 motion segments can be specified in one motion block. In total, the module can hold up to 114 motion segments per axis.
50084

Figure 9.2 illustrates a motion profile consisting of five motion segments. Segments 14 through 17 move the axis in one direction, while segment 18 returns it to its original position and triggers segment 14. The solid line indicates the actual axis movement and the dotted lines show the profile of each motion segment if its motion were not interrupted by the triggering of the subsequent motion segment. See Chapter 10 for a detailed explanation of the ladder logic for this example (Sample Application Program #2).

Figure 9.2
Profile Using Motion Segments

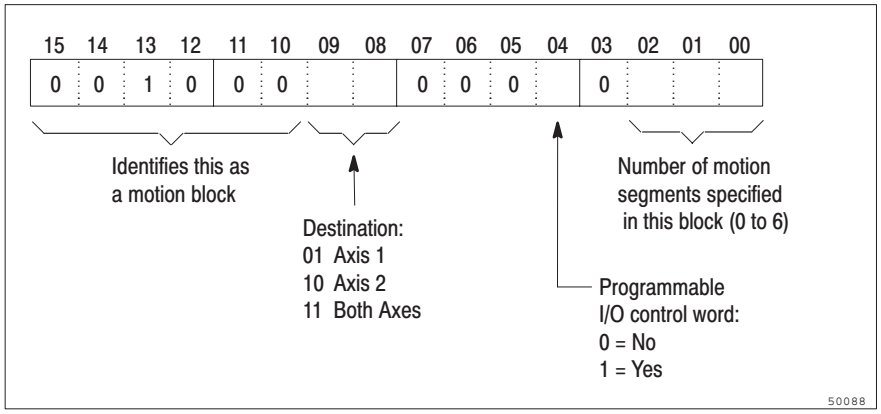


Important: When linking multiple motion segments together, as shown in Figure 9.2, it is important to remember that while the axis is in motion, the deceleration cannot be decreased until after the done bit goes high and deceleration can only be increased if velocity smoothing is disabled. The same criteria apply as for issuing software start commands with the axis in motion. See the section, *Command Block*, in Chapter 7.

Motion Block Control Word

The motion block control word identifies the block as a motion block, specifies the number of motion segments defined in the block, and indicates whether or not it contains a programmable I/O control word. (See Figure 9.3.) Nineteen motion blocks must be downloaded to the module to configure all 114 motion segments.

Figure 9.3
Motion Block Control Word


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 1 0 0 0 0 0 0 0 0
Identifies this as a motion block
Destination:
01 Axis 1
10 Axis 2
11 Both Axes
Number of motion segments specified in this block (0 to 6)
Programmable I/O control word:
0 = No
1 = Yes
50088

Bit 4 - Programmable I/O Word

If you want to configure the programmable I/O, set this bit to 1 to indicate to the module that a programmable I/O control word is appended to the end of the motion block. If you do not specify I/O control information, a default configuration is used.

Bit 8 - Axis 1

Set this bit to 1 if the motion block applies to axis 1.

Bit 9 - Axis 2

Set this bit to 1 if the motion block applies to axis 2.

Important: If bits 8 and 9 are both set, the motion block applies to both axes and a single block transfer of a new motion block will update the module's internal motion segment data table for both axes.

Programmable Input and Output

You can configure the general purpose inputs, INPUT 1 and/or INPUT 2 so that, given their state and the trigger conditions in the current segment, another motion segment may be triggered. Also, any motion segment may optionally pulse or latch the programmable outputs, OUTPUT 1 and OUTPUT 2, when the trigger conditions are satisfied.

Programmable I/O Control Word

With the programmable I/O word you can specify the level/edge sensitivity triggers for INPUT 1 and INPUT 2 and specify programmable output parameters for OUTPUT 1 and OUTPUT 2. (See Figure 9.4.) At any given time, only one programmable I/O configuration per axis is valid. As soon as a programmable I/O control word is received by the module, it overwrites the previous I/O configuration.

Bits 0 through 10, except for bits 2 and 6, define the configuration of OUTPUT 1 and OUTPUT 2. Bits 2, 6 and 11 are reserved, while bits 12 through 15 define the trigger configuration for INPUT 1 and INPUT 2. If OUTPUT 1 and OUTPUT 2 are defined as in-position and loop fault outputs, any relevant configuration specified by bits 0 through 8 are ignored by the module.

Bits 0 and 1 - OUTPUT 1 Pulse Duration

If OUTPUT 1 is configured to be programmable and motion segment control word 2 specifies that OUTPUT 1 is pulsed when trigger conditions are satisfied, these bits define the duration of the pulse to be 20, 50, 100 or 200 milliseconds.

Bit 3 - Normal/Complement OUTPUT 1

If OUTPUT 1 is configured to be programmable, this bit defines whether OUTPUT 1 is normal (active high) or complement (active low).

Normal programmable outputs go or stay:

- high when triggered to latch on
- low when triggered to latch off
- high for the specified duration when triggered to pulse

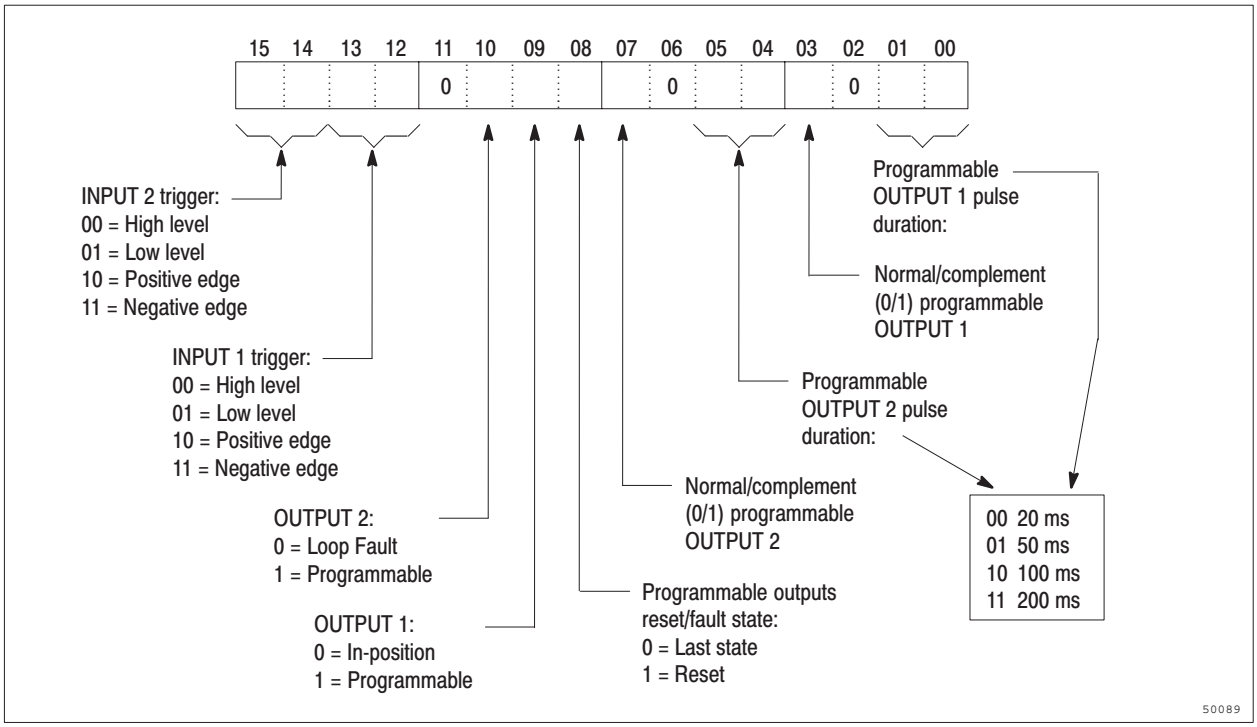
Complement programmable outputs go or stay:

- low when triggered to latch on
- high when triggered to latch off

- low for the specified duration when triggered to pulse

When an output changes to a high or low state, it is guaranteed to stay in that state for a minimum of 16 milliseconds in order to be compatible with the discrete inputs. Thus if two successive changes to an output are initiated within 16 milliseconds of each other, the second change will be delayed until the 16 millisecond interval has expired. If the second change is a pulse, the duration of that pulse will also be shortened by the length of the additional delay.

Figure 9.4
Programmable I/O Control Word



Bits 4 and 5 - OUTPUT 2 Pulse Duration

If OUTPUT 2 is configured to be programmable and motion segment control word 2 specifies that OUTPUT 2 is pulsed when trigger conditions are satisfied, these bits define the duration of the pulse to be 20, 50, 100 or 200 milliseconds.

Bit 7 - Normal/Complement OUTPUT 2

If OUTPUT 2 is configured to be programmable, this bit defines whether OUTPUT 2 is normal (active high) or complement (active low). See the description for bit 3 of this word.

Bit 8 - Outputs Reset/Fault State

This bit indicates whether OUTPUT 1 and OUTPUT 2 (when configured as programmable outputs) remain in their last state or are reset when:

- a software reset command is sent to the module
- the module has faulted
- the programmable controller has faulted
- the programmable controller is placed in programming mode

Module faults include loss of analog power, analog interface fault, memory fault, discrete input fault, transducer interface fault, excess following error, excess PID error, loss of feedback, hardware stop input or an immediate stop command.

Bit 9 - OUTPUT 1

You set this bit to 1 to indicate that OUTPUT 1 is programmable. If you clear this bit, OUTPUT 1 defaults to an in-position output.

Bit 10 - OUTPUT 2

You set this bit to 1 to indicate that OUTPUT 2 is programmable. If you clear this bit, OUTPUT 2 defaults to a loop fault output.

Bits 12, 13 - INPUT 1 Trigger

With these bits you can set the level/edge trigger sensitivity for INPUT 1 to high level, low level, positive edge or negative edge.

Bits 14, 15 - INPUT 2 Trigger

With these bits you can set the level/edge trigger sensitivity for INPUT 2 to high level, low level, positive edge or negative edge.

Default I/O Configuration

If you do not download the programmable I/O control word, the module defaults both axes to:

INPUT 1	positive edge trigger
INPUT 2	high level trigger
OUTPUT 1	in-position output
OUTPUT 2	loop fault output

Motion Segments

A motion segment consists of a setpoint, trigger conditions, and programmable output options. (See Figure 9.2.)

Motion Segment Control Words

The motion segment control words (see Figure 9.5) contain the current motion segment ID number, trigger conditions, the next motion segment ID number, and programmable output specifications. When the trigger conditions are satisfied, the motion segment identified by the next motion segment ID is executed and the programmable outputs are updated based on the trigger output and output control bits.

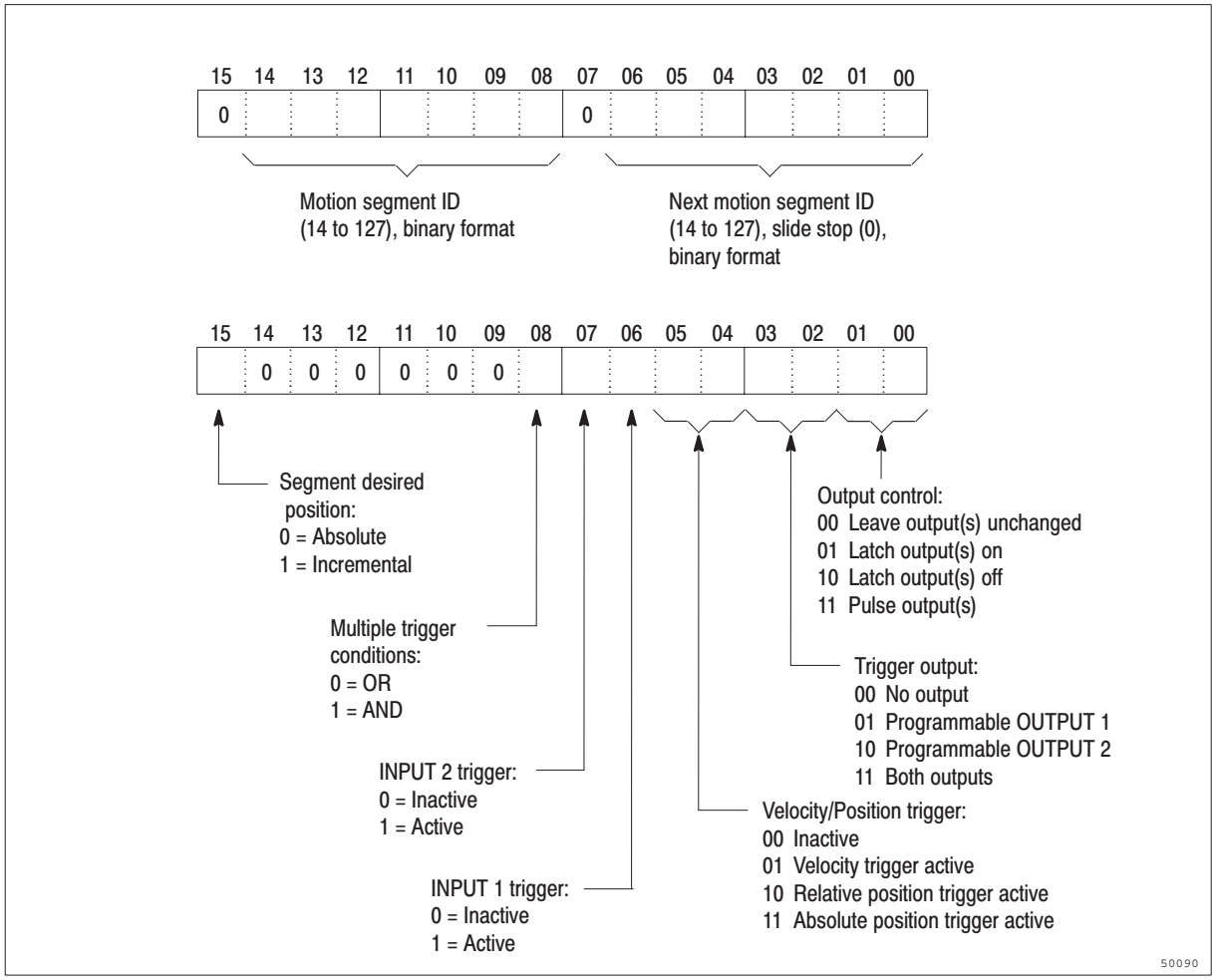
Control Word 1: Bits 0 to 6 - Next Motion Segment ID

These bits identify the next motion segment to be executed when this motion segment's trigger conditions are satisfied. In order for the next motion segment to be successfully executed, you must ensure that the motion segment referenced by the next motion segment ID is defined. If the next motion segment ID is set to 0, it causes a slide stop when the trigger conditions are satisfied.

Control Word 1: Bits 8 to 14 - Motion Segment ID

These bits contain a unique identification number between 14 and 127 (selected by the user), in binary format, for the current motion segment. This allows it to be referenced by another motion segment.

Figure 9.5
Motion Segment Control Words



Control Word 2: Bits 0 and 1 - Output Control

These two bits specify whether the output(s) specified in bits 2 and 3 should be unchanged, latched on, latched off, or pulsed, when this motion segment's trigger conditions are satisfied.

Control Word 2: Bits 2 and 3 - Trigger Output

These bits specify the programmable outputs (OUTPUT 1, OUTPUT 2, both or neither) that will be affected when this motion segment's trigger conditions are satisfied.

Control Word 2: Bits 4 and 5 - Velocity/Position Trigger

These bits indicate if one of the velocity, relative position, or absolute position triggers is active. If the velocity trigger is active, you must specify in the trigger velocity/position word, the absolute velocity at which the trigger condition will be satisfied. If the relative position trigger is active, the trigger position you specify in the trigger velocity/position words is relative to the current axis position when the current motion segment began execution. If the absolute position trigger is active you must specify an absolute position in the trigger velocity/position words.

Important: Since it is unlikely that the exact velocity/position of the axis will precisely equal the trigger velocity/position specified, the trigger condition is satisfied when the actual velocity/position reaches or crosses the trigger velocity/position.

Control Word 2: Bits 6 and 7 - INPUT 1 and INPUT 2 Triggers

These bits indicate if INPUT 1 and/or INPUT 2 triggers are active. If an input trigger is active, it means that the associated discrete input must be satisfied, based on trigger sensitivity, to cause the execution of the next motion segment. The trigger sensitivity is specified in the programmable I/O control word (bits 12 to 15), and can be high/low level or positive/negative edge.

Control Word 2: Bit 8 - Multiple Trigger Conditions

If more than one trigger condition is active, this bit controls whether all the specified active trigger conditions (AND) or just one (OR) must be satisfied. If you specify AND, the active trigger conditions do not necessarily have to occur at the same time to trigger the next motion segment. The active trigger conditions can occur in sequence.

Control Word 2: Bit 15 - Desired Position

This bit specifies that the movement profile’s desired position given in this motion segment is incremental (1) or absolute (0).

Important: Absolute moves define a new endpoint in relation to the zero-position offset specified in the parameter block. Incremental moves define a new endpoint in relation to the current axis position at the start of the current motion segment’s execution.

Desired Position, Local Velocity, Local Acceleration and Local Deceleration Words

The format of the (MS) desired position, (LS) desired position, local velocity, local acceleration and local deceleration words in the motion block (see Figure 9.1) is the same as the format of the (MS) setpoint position, (LS) setpoint position, local velocity and local deceleration words in the setpoint block.

Trigger Velocity/Position Words

If you specify a position or velocity trigger, in control word 2 you must specify a corresponding position or velocity in the trigger velocity/position words. If neither a velocity nor position trigger is specified in control word 2, the trigger velocity/position words are ignored. The data format for the position and velocity words is the same as the format for the (MS) setpoint position, (LS) setpoint position and local velocity, in the setpoint block.

Important: The module interprets all velocities as magnitudes without positive or negative direction. Therefore, a velocity trigger condition will be satisfied when the current velocity magnitude (axis moving in either direction) crosses the specified trigger velocity magnitude.

The Command Block and the Motion Block

Bits 0 to 6 of axis control word 2 in the command block are used to identify the next setpoint or motion block you want to command. To initiate a motion block movement, you command one of its motion segments by placing the motion segment ID in bits 0 to 6. A motion segment ID number can be from 14 to 127 in binary format. (See Figure 7.45.)

The Status Block and the Motion Block

Status word 2 will indicate when motion segments are active (see Figure 6.4) while the extended status words 10 and 11 will contain the ID number of the active motion segment (see Figure 6.8). Any programming errors in the motion blocks or any operational errors that occur while using motion segments will be reported in words 4, 5 and 8, 9 of the status block (see Figure 6.5 and Table 6.A) if those words are configured by the command block to return diagnostics.

Using the Motion Block

As mentioned previously, because initiating a single motion segment from the command block can trigger a sequence of motions, you must exercise caution when using the motion block. For safety reasons, a watchdog monitors the state of the programmable controller. If it faults or enters programming mode while a motion segment is executing, the watchdog in the module initiates a slide stop. The programmable outputs in the motion block will be set to either their last state or to reset state, depending on how bit 8 (programmable outputs reset/fault state) in the programmable I/O word is configured.

Important: When the module is reset with the command block, previously defined motion segment and setpoint data is lost, but the programmable I/O configuration remains as it was prior to axis reset. If power is cycled, the programmable I/O configuration changes to its default.

Here are some recommended ways of stopping axis motion while motion segments are active. The one you use will depend upon your particular application.

- perform a slide stop (software)
- execute an immediate stop (hardware or software)
- initiate a setpoint (the final axis position will be the setpoint's endpoint)
- initiate a motion segment which will not have its trigger conditions satisfied (the final axis position will be at the motion segment's endpoint)
- put the programmable controller in programming mode

Important: When a motion segment is commanded with the hardware start enabled, hardware starts are automatically accepted during axis motion regardless of the Stop/Start enhancement setting (axis control word 1 in the command block). When you initiate a motion segment with the hardware start enabled in the command block, the specified motion segment will not begin execution until the module recognizes a rising edge on the hardware start input. However, subsequent motion segments triggered by another motion segment will not require a hardware start to begin execution.

Important: Incremental motion segments and relative position triggers are based on the current axis position at the beginning of the motion segment's execution. Because of this, if you link a series of incremental motion segments together you will likely see a small build-up of error. It is recommended that an absolute position move be done occasionally to remove this error build-up. Also, because incremental setpoint position is based on the previous setpoint's endpoint, but incremental motion segment position is based on the current axis position at the beginning of the motion segment's execution, it is recommended that you not use a combination of incremental setpoints and motion segments together.

Chapter 10

Sample Application Programs

This chapter gives a general explanation of how to program programmable logic controllers and provides the code for, and descriptions of, the two sample application programs contained on the disk that accompanies the Linear Positioning Module. Application Program 1 shows how to implement axis movement for a single axis using a setpoint block containing four setpoints. Application Program 2 shows how to implement axis movement for a single axis using the motion block discussed in Chapter 9. The application programs on the accompanying disk have been developed for both the 6200 and ICOM software packages. The first application program is called QB1 and the second is QB2. Another application program, QB2_515A, should be used if you are running a Series A, PLC-5/15.

Programming Objectives

The main objectives of a programmable controller program for the Linear Positioning Module are:

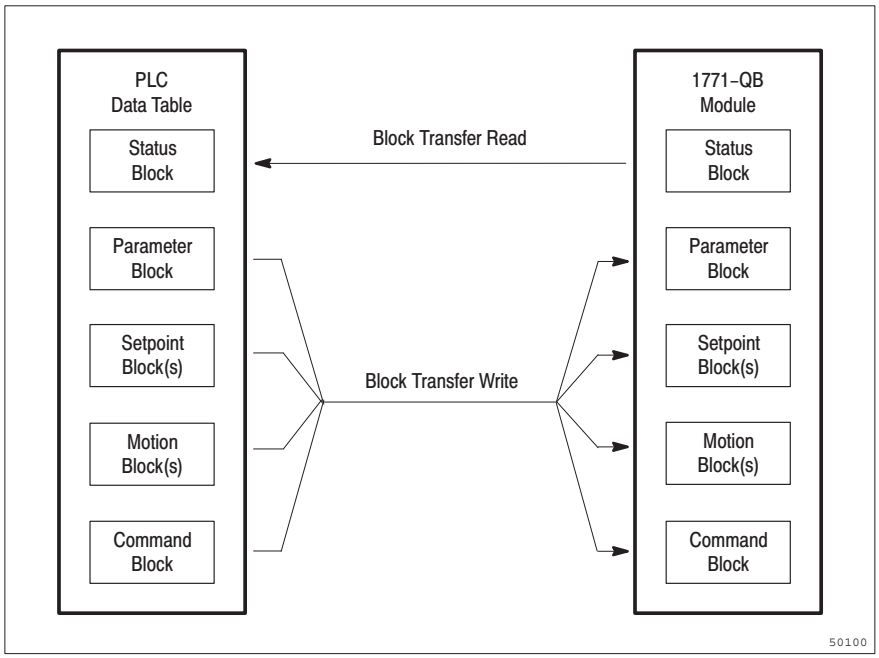
- to regularly transfer the status block from the Linear Positioning Module to the programmable logic controller's data tables. You achieve this through a block read instruction.
- to transfer the parameter block, the command block, the setpoint block (optional), and the motion block (optional) from the programmable controller's data table to the Linear Positioning Module. The program must manipulate the source address of the block transfer write instruction to determine which block to send next.

After powerup or axis reset, the program must send data blocks to the module in the following order:

1. Parameter block
2. Setpoint block(s) - if you only command setpoint 13 or a motion segment in the command block, you won't need to send a setpoint block
3. Motion block(s) (if required)
4. Command block

The program will then continue sending command blocks during normal operation. The module does not request the blocks in the required order; the program must maintain this order.

Figure 10.1
Overview of Block Transfers



Block Transfer Sequencing

To ensure correct block transfer sequence on powerup, it is recommended that the programs:

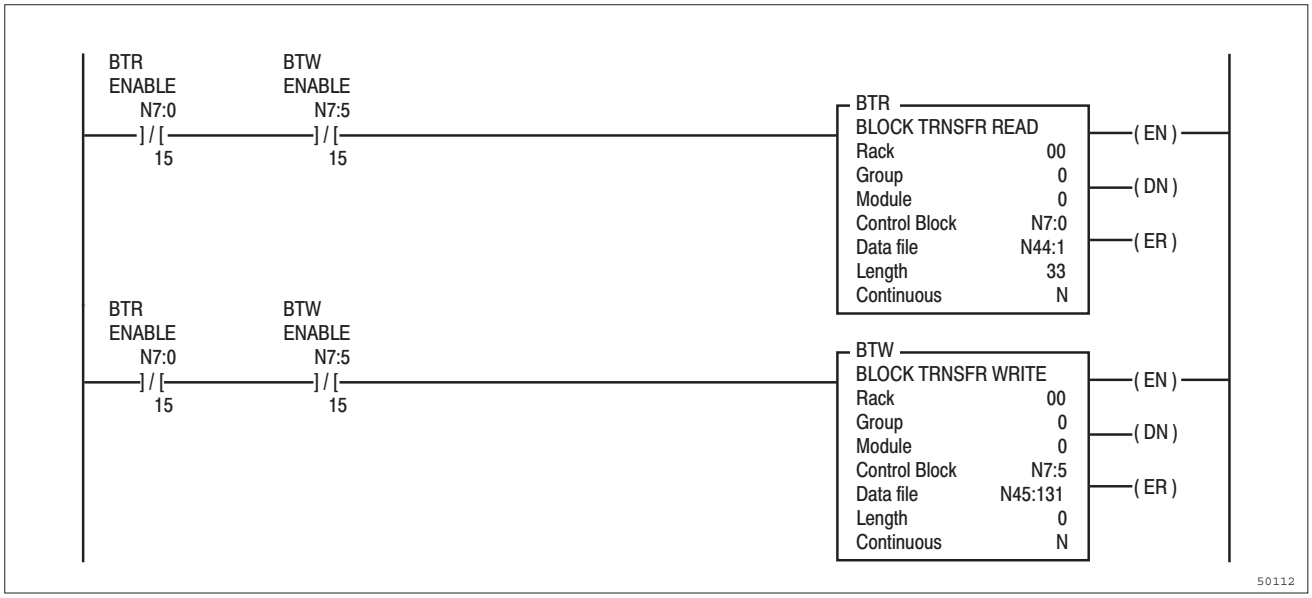
- make parameter block transfer conditional on the status block's ready bit being low
- make setpoint block transfer conditional on the status block's setpoints received bit for this control loop being low and on the ready bit being high (if you're using setpoint blocks)
- make the next motion block transfer conditional on the block transfer toggle bit changing from the state it was in prior to the transfer of the previous block
- make command block transfer conditional on the setpoints received bit and the ready bit both being high
- if you aren't using setpoint blocks, make command block transfer conditional on just the ready bit being high

PLC-5 Block Transfer Instructions

You should program a PLC-5 processor's block transfer to use the bidirectional method to avoid problems when troubleshooting the module. However, block transfer writes only need to be enabled when a command block, motion blocks, setpoint blocks or a parameter block must be sent to the module.

Important: Processor execution of block transfer instructions is asynchronous to the program scan. The status of these bits could change at any point in the program scan. If the data must be synchronized to the program scan, transfer the block transfer read instruction to a storage location via a file-to-file move. The program can then examine data in the storage location.

Figure 10.2
Block Transfer Instructions for PLC-5 Controllers



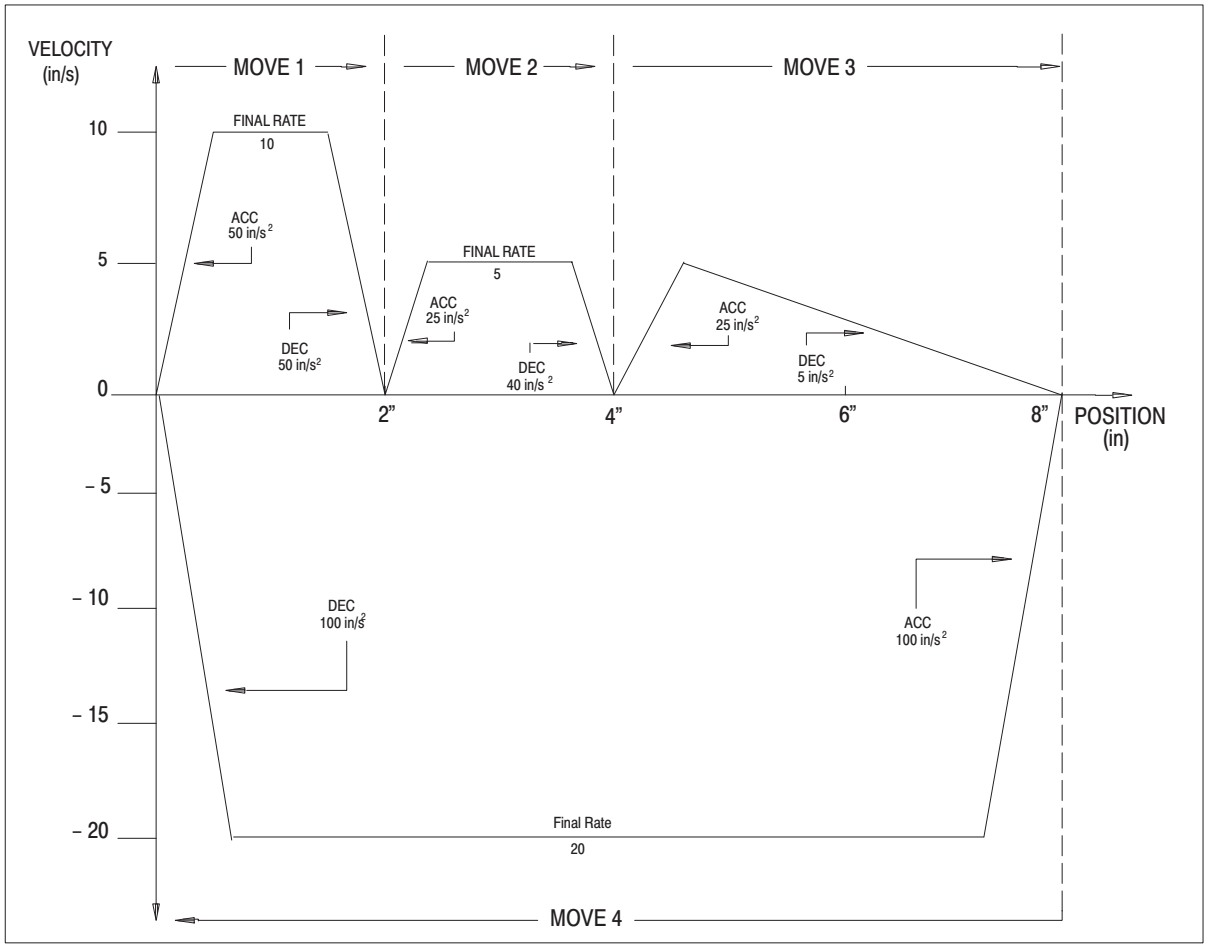
Application Program #1

With this simple application program, the module controls the motion of a single axis. Figure 10.3 shows the axis movement profile. Four setpoints in the setpoint block control axis motion for this loop. The first three setpoints specify local velocity and acceleration/deceleration, while the fourth setpoint uses the global velocity and acceleration/deceleration defined in the parameter block. Setting the fourth setpoint's local acceleration and deceleration words to zero forces it to use global parameters. The first three setpoints cause axis movement in one direction, while the fourth returns the axis to the original position.

Important: Note that:

- the program doesn't issue the start command for each move until after the module reports in-position (through the status block) from the previous move
- due to the specified deceleration rate of move 3, the axis will not achieve the final rate of 5 in/s

Figure 10.3
Movement Profile for Application Program #1



Planning the Data Blocks for Application Program #1

For this example, we assume a PLC-5/15 controller and assign the data blocks shown in Table 10.A. The files used are shown in Table 10.B.

Table 10.A
Data Blocks for Application Program #1

Block	Starting Address
Status	N44:1
Parameter	N45:1
Setpoint	N45:61
Command	N45:131

Table 10.B
Files for Application Program #1

File	Size (Elements)	Usage
R6	1	sequencer control
N7	10	block transfer control
D9	5	sequencer data
N44	34	status block
N45	145	parameter, setpoint, command blocks

Figure 10.4 to Figure 10.7 show the hexadecimal values for the parameter, setpoint and command blocks, and necessary sequencer data for this example.

Figure 10.4
Data Table Contents for Application Program #1 - Parameter Block

Project Name: Application Program #1 - Axis 1 Page____ of ____
Designer: _____ Address ____ of ____
Date:_____ Axis No. 1 Block Description: Parameter

Data Table Address	Position	File Data				Description
N45:1	**1**	4	1	0	9	Parameter control word (Axis 1, inches, BCD)
2	**2**	0	1	0	0	Analog range (100%)
3	**3**	2	5	0	0	+Analog calibration constant. (25.00 ips)
4	**4**	2	5	0	0	–Analog calibration constant (25.00 ips)
5	**5**	0	0	0	9	(MS) Transducer calibration constant (9.0500 microsec/inch)
6	**6**	0	5	0	0	(LS) Transducer calibration constant
7	**7**	8	0	0	5	(MS) Zero-position offset (-5.000 inches)
8	**8**	0	0	0	0	(LS) Zero-position offset
9	**9**	0	2	1	0	+Software travel limit (21.0 inches)
10	**10**	8	0	1	0	–Software travel limit (-1.0 inch)
11	**11**	0	1	0	0	In-position band (0.100 inches)
12	**12**	0	5	0	0	PID band (0.500 inches)
13	**13**	0	0	0	0	Deadband (0.000 inches)
14	**14**	1	0	0	0	Excess following error (1.000 inch)
15	**15**	1	0	0	0	Maximum PID error (1.000 inch)
16	**16**	0	0	1	0	Integral term limit (10%)
17	**17**	0	6	0	0	Proportional gain (0.0600 ips/mil)
18	**18**	0	0	0	0	Gain break speed (0.00 ips)
19	**19**	0	0	0	0	Gain factor (0.00)
20	**20**	0	4	2	0	Integral gain (0.0420 (ips/s)/mil)
21	**21**	0	4	2	0	Derivative gain (0.0420)
22	**22**	0	3	0	0	Feedforward gain (30.0%)
23	**23**	1	0	0	0	Global velocity (10.00 ips)
24	**24**	1	0	0	0	Global acceleration (100.0 ips/s)
25	**25**	1	0	0	0	Global deceleration (100.0 ips/s)
26	**26**	0	3	0	0	Velocity smoothing constant (3.00 (ips/s)/ms)
27	**27**	0	1	0	0	Low jog rate (1.00 ips)
28	**28**	1	0	0	0	High jog rate (10.00 ips)
29	**29**	0	0	0	0	Reserved
30	**30**	0	0	0	0	Reserved

Figure 10.5
Data Table Contents for Application Program #1 - Setpoint Block

Project Name: Application Program #1 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Setpoint

Data Table Address	Position	File Data				Description
N45:61	**1**	8	1	0	4	Setpoint block control word (Axis 1, 4 setpoints)
62	**2**	0	0	0	0	Incremental/absolute word (all absolute)
63	**3**	0	0	0	2	Move #1 (MS) Setpoint position (2.000 inches)
64	**4**	0	0	0	0	(LS) Setpoint position
65	**5**	1	0	0	0	Local velocity (10.00 ips)
66	**6**	0	5	0	0	Local acceleration (50.0 ips/s)
67	**7**	0	5	0	0	Local deceleration (50.0 ips/s)
68	**8**	0	0	0	4	Move #2 (MS) Setpoint position (4.000 inches)
69	**9**	0	0	0	0	(LS) Setpoint position
70	**10**	0	5	0	0	Local velocity (5.00 ips)
71	**11**	0	2	5	0	Local acceleration (25.0 ips/s)
72	**12**	0	4	0	0	Local deceleration (40.0 ips/s)
73	**13**	0	0	0	8	Move #3 (MS) Setpoint position (8.000 inches)
74	**14**	0	0	0	0	(LS) Setpoint position
75	**15**	0	5	0	0	Local velocity (5.00 ips)
76	**16**	0	2	5	0	Local acceleration (25.0 ips/s)
77	**17**	0	0	5	0	Local deceleration (5.0 ips/s)
78	**18**	0	0	0	0	Move #4 (MS) Setpoint position (0.000 inches)
79	**19**	0	0	0	0	(LS) Setpoint position
80	**20**	2	0	0	0	Local velocity (20.00 ips)
81	**21**	0	0	0	0	Local acceleration (global acceleration)
82	**22**	0	0	0	0	Local deceleration (global deceleration)

Figure 10.6
Data Table Contents for Application Program #1 - Command Block

Project Name: Application Progam #1 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Command

Data Table Address	Position	File Data				Description
N45: 131	**1**	E	0	0	4	Axis control word 1 (Diagnostic, auto)
132	**2**	0	0	0	0	Axis control word 2
133	**3**	0	0	0	0	(MS) Setpoint 13 position
134	**4**	0	0	0	0	(LS) Setpoint 13 position
135	**5**	0	0	0	0	Setpoint 13 velocity
136	**6**	0	0	0	0	Setpoint 13 acceleration
137	**7**	0	0	0	0	Setpoint 13 deceleration

Figure 10.7
Data Table Contents for Application Program #1 - Sequencer Data

Project Name: Application Program #1 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. N/A Block Description: Sequencer Data

Data Table Address	Position	File Data				Description
D9: 0	**1**	0	0	0	0	Axis 1 setpoint sequence
1	**2**	0	0	0	1	
2	**3**	0	0	0	2	
3	**4**	0	0	0	3	
4	**5**	0	0	0	4	

Program Rungs for Application Program #1

Figure 10.8 shows the ladder diagram programming for this application on a PLC-5/15 system. The rungs perform the following functions:

Rung 2:0

Rung 2:0 reads the module's status block and, in conjunction with rung 2:4, performs bidirectional block transfers to and from the module.

Rung 2:1

Rungs 2:1, 2:2, and 2:3 determine which block (parameter, setpoint, or command) will be sent to the module via the next block transfer write (BTW). If the axis 1 ready bit is low, (the module is in powerup or a reset command has just been executed), rung 2:1 moves the source address of the parameter block into the BTW's control block. This causes the programmable controller to send the parameter block when rung 2:4 is executed.

Rung 2:2

If the ready bit is high and the setpoints received bit is low (the module has received a valid parameter block but has not yet received a setpoint block), rung 2:2 moves the source address of the setpoint block into the BTW's control block. This causes the programmable controller to send the setpoint block when rung 2:4 is executed.

Rung 2:3

If the ready bit and the setpoints received bit are high, (the module has received a valid parameter block and the setpoint block), rung 2:3 moves the source address of the command block into the BTW's control block. This causes the programmable controller to send the command block when rung 2:4 is executed.

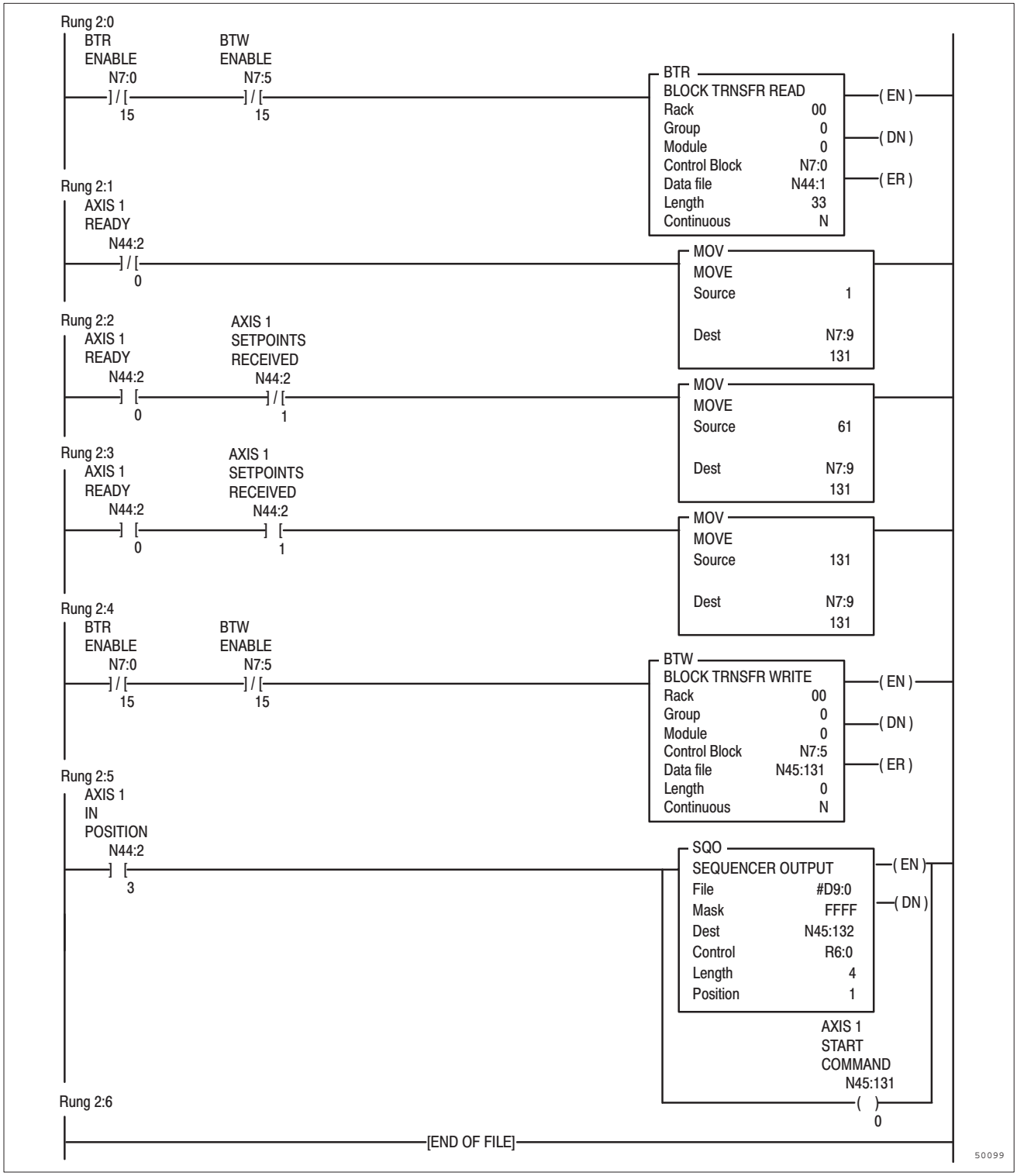
Rung 2:4

Rung 2:4 writes the block selected by rung 2:1, 2:2, or 2:3 to the module.

Rung 2:5

Rung 2:5 controls the order of axis 1 moves and issues the start command for the move. When axis 1's in-position bit goes high, the sequencer alters control word 2 in the command block to indicate the next setpoint and the rung issues the start command. When the next BTW occurs (sends the altered command block), the next setpoint movement profile begins executing and the in-position bit goes low again.

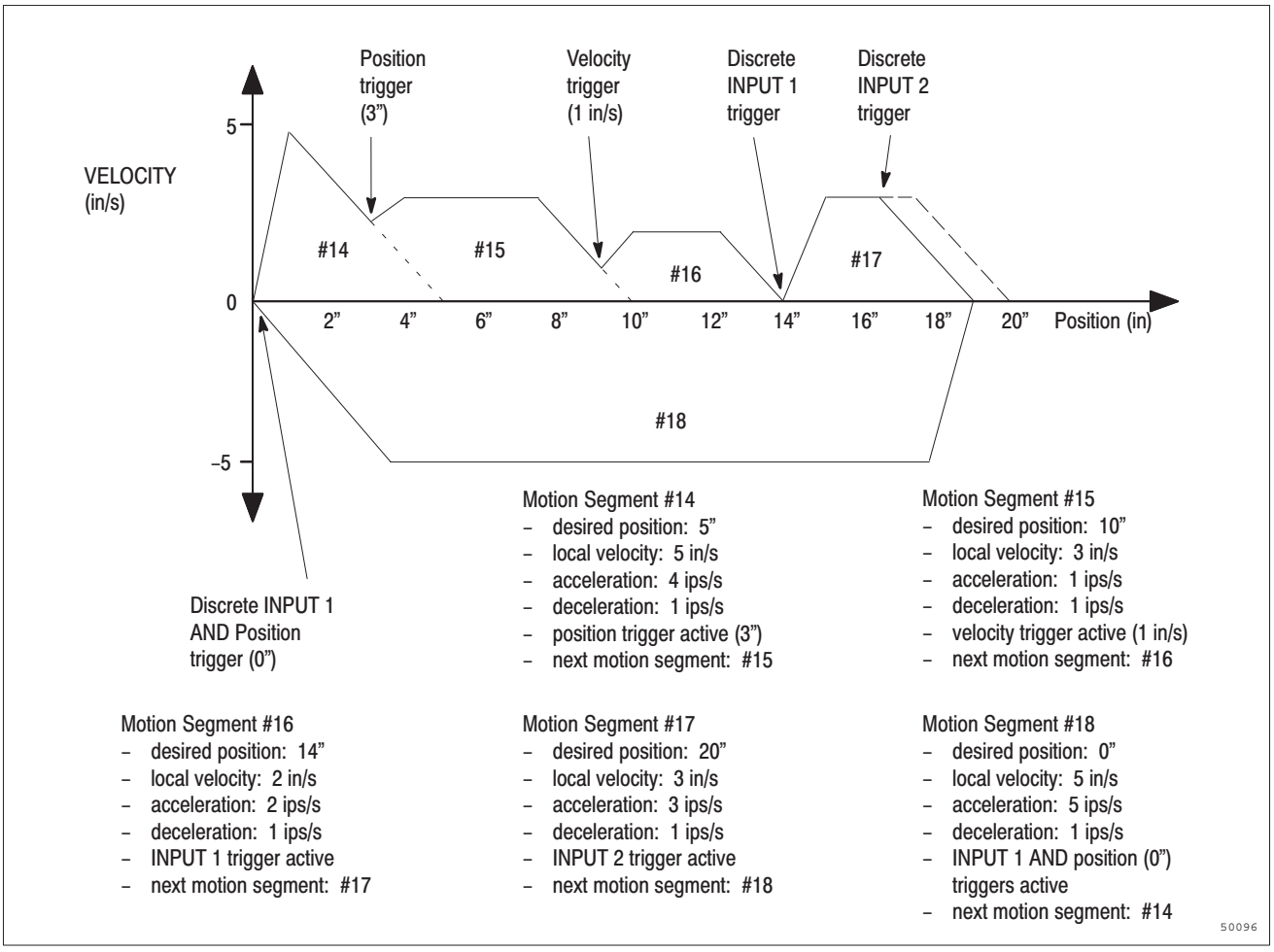
Figure 10.8
Program Rungs for Application Program #1



Application Program #2

This application program illustrates how to use a module to control the motion of a single axis using motion blocks. (See Chapter 9.) Figure 10.9 shows the movement profile for this program. Five motion segments describe axis movement. Segments 14 through 17 move the axis in one direction, and segment 18 returns it to its original position and triggers the first motion segment (#14). The solid movement profile line indicates the actual axis movement. The dotted profile lines show the movement profiles of each motion segment if its motion were not interrupted by the triggering of the subsequent movement profile. When motion segment 14's trigger conditions are satisfied, programmable OUTPUT 1 is pulsed for 200 ms. When motion segment 17's trigger conditions are satisfied, programmable OUTPUT 2 is latched high, and when motion segment 18's trigger conditions are satisfied, programmable OUTPUT 2 is latched low.

Figure 10.9
Axis Profile for Motion Block



Important: Note that:

- due to the specified acceleration and deceleration rate of move #14, the axis will not achieve the final velocity rate of 5 in/s
- because moves #16 and #17 have discrete input triggers which may be triggered at any time during their movements profiles, the axis may not achieve the final velocity rates of 2 in/s and 3 in/s
- all motion segments and programmable I/O information could be contained in a single motion block. Three motion blocks are used here to demonstrate the recommended method of loading multiple motion blocks into a module.

Planning the Data Blocks for Application Program #2

For application program 2 a PLC-5/15 processor is assumed and the data blocks are assigned as shown in Table 10.C. The files used are shown in Table 10.D. The data table for the parameter block for application program 2 is the same as shown in Figure 10.4, for application program 1.

Table 10.C
Data Blocks for Application Program # 2

Block	Starting Address
Status	N44:1
Parameter	N45:1
Command	N45:131
Motion	N80:1

Table 10.D
Files for Application Program #2

File	Size (Elements)	Usage
B3	2	last block transfer toggle control
R6	3	sequencer control
N7	10	block transfer control
D9	20	sequencer data
N44	34	status block
N45	145	parameter, command blocks
N80	53	motion block

Figure 10.10 to Figure 10.14 show the hexadecimal values for the motion and command blocks, and necessary sequencer data for this example.

Figure 10.10
Data Table Contents for Application Program #2 - Motion Block 1

Project Name: Application Program #2 - Axis 1 Page____ of ____
Designer: _____ Address ____ of ____
Date:_____ Axis No. 1 Block Description: Motion Block 1

Data Table Address	Position	File Data				Description
N80:1	**1**	2	1	0	3	Motion block control word (3 motion segments)
2	**2**	0	E	0	F	#14 Motion segment control word 1 (next motion segment: #15)
3	**3**	0	0	3	7	Motion segment control word 2 (absolute position trigger)
4	**4**	0	0	0	5	(MS) Desired position (5.000 inches)
5	**5**	0	0	0	0	(LS) Desired position
6	**6**	0	5	0	0	Local velocity (5.00 ips)
7	**7**	0	0	4	0	Local acceleration (4.0 ips/s)
8	**8**	0	0	1	0	Local deceleration (1.0 ips/s)
9	**9**	0	0	0	3	Trigger velocity or (MS) trigger position (3.000 inches)
10	**10**	0	0	0	0	(LS) Trigger position
11	**11**	0	F	1	0	#15 Motion segment control word 1 (next motion segment: #16)
12	**12**	0	0	1	0	Motion segment control word 2 (velocity trigger)
13	**13**	0	0	1	0	(MS) Desired position (10.000 inches)
14	**14**	0	0	0	0	(LS) Desired position
15	**15**	0	3	0	0	Local velocity (3.00 ips)
16	**16**	0	0	1	0	Local acceleration (1.0 ips/s)
17	**17**	0	0	1	0	Local deceleration (1.0 ips/s)
18	**18**	0	1	0	0	Trigger velocity or (MS) trigger position (1.00 ips)
19	**19**	0	0	0	0	(LS) Trigger position
20	**20**	1	0	1	1	#16 Motion segment control word 1 (next motion segment: #17)
21	**21**	0	0	4	0	Motion segment control word 2 (input 1 trigger)
22	**22**	0	0	1	4	(MS) Desired position (14.000 inches)
23	**23**	0	0	0	0	(LS) Desired position
24	**24**	0	2	0	0	Local velocity (2.00 ips)
25	**25**	0	0	2	0	Local acceleration (2.0 ips/s)
26	**26**	0	0	1	0	Local deceleration (1.0 ips/s)
27	**27**	0	0	0	0	Trigger velocity or (MS) trigger position
28	**28**	0	0	0	0	(LS) Trigger position

Figure 10.11
Data Table Contents for Application Program #2 - Motion Block 2

Project Name: Application Program #2 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Motion Block 2

Data Table Address	Position	File Data				Description
N80:31	**1**	2	1	0	2	Motion block control word (2 motion segments)
32	**2**	1	1	1	2	#17 Motion segment control word 1 (next motion segment: #18)
33	**3**	0	0	8	9	Motion segment control word 2 (input 2 trigger)
34	**4**	0	0	2	0	(MS) Desired position (20.000 inches)
35	**5**	0	0	0	0	(LS) Desired position
36	**6**	0	3	0	0	Local velocity (3.00 ips)
37	**7**	0	0	3	0	Local acceleration (3.0 ips/s)
38	**8**	0	0	1	0	Local deceleration (1.0 ips/s)
39	**9**	0	0	0	0	Trigger velocity or (MS) trigger position
40	**10**	0	0	0	0	(LS) Trigger position
41	**11**	1	2	0	E	#18 Motion segment control word 1 (next motion segment #14)
42	**12**	0	1	7	A	Motion segment control word 2 (input 1 AND position trigger)
43	**13**	0	0	0	0	(MS) Desired position (0.000 inches)
44	**14**	0	0	0	0	(LS) Desired position
45	**15**	0	5	0	0	Local velocity (5.00 ips)
46	**16**	0	0	5	0	Local acceleration (5.0 ips/s)
47	**17**	0	0	1	0	Local deceleration (1.0 ips/s)
48	**18**	0	0	0	0	Trigger velocity or (MS) trigger position (0.000 inches)
49	**19**	0	0	0	0	(LS) Trigger position

Figure 10.12
Data Table Contents for Application Program #2 - Motion Block 3

Project Name: Application Program #2 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Motion Block 3

Data Table Address	Position	File Data				Description
N80:51	**1**	2	1	1	0	Motion block control word (programmable I/O word appended)
52	**2**	0	6	3	3	Programmable I/O control word (outputs 1 and 2 programmable)

Figure 10.13
Data Table Contents for Application Program #2 - Command Block

Project Name: Application Progam #2 – Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Command

Data Table Address	Position	File Data				Description
N45:131	**1**	E	0	0	5	Axis control word 1 (diagnostic, auto, start)
132	**2**	0	0	1	0	Axis control word 2 (motion segment #16)
133	**3**	0	0	0	0	(MS) Setpoint 13 position
134	**4**	0	0	0	0	(LS) Setpoint 13 position
135	**5**	0	0	0	0	Setpoint 13 velocity
136	**6**	0	0	0	0	Setpoint 13 acceleration
137	**7**	0	0	0	0	Setpoint 13 deceleration

Figure 10.14
Data Table Contents for Application Program #2 - Sequencer Data

Project Name: Application Program #2 - Axis 1 Page ____ of ____
Designer: _____ Address ____ of ____
Date: _____ Axis No. 1 Block Description: Sequencer Data

Data Table Address	Position	File Data				Description
D9:0	**1**	0	0	0	0	Axis 1 Motion blocks to be loaded file address sequence
1	**2**	0	0	5	0	
2	**3**	0	0	5	0	
3	**4**	0	0	5	0	
4	**5**	0	0	5	0	
5	**6**					
6	**7**					
7	**8**					
8	**9**					
9	**10**					
10	**11**	0	0	0	0	Axis 1 Motion blocks to be loaded word address sequence
11	**12**	0	0	0	1	
12	**13**	0	0	1	F	
13	**14**	0	0	3	3	
14	**15**	0	0	3	3	

Program Rungs for Application Program #2

Figure 10.15 and Figure 10.16 show the ladder diagram programming for this application on a PLC-5/15 system. The rungs perform the following functions:

Rung 2:0

Rung 2:0 reads the module's status block and, in conjunction with rung 2:5, performs bidirectional block transfers to and from the module.

Important: Set the SQO length in rung 2:3 to the number of motion blocks to be loaded, plus 1.

Rung 2:1

Rungs 2:1, 2:3 and 2:4 determine which block (parameter, motion or command) will be sent to the module via the next block transfer write (BTW). If the axis 1 ready bit is low (the module is in powerup or a reset has just been executed), rung 2:1 moves the source address of the parameter block into the BTW's control block, causing the programmable controller to send the parameter block when rung 2:5 is executed.

Rung 2:2

When the ready bit first goes high (the module has just received a valid parameter block), the control files for the sequencer output instructions (SQO) are reset and a NOT instruction is performed on the status block word containing the block transfer toggle bit (bit 7). The result is stored in reference word B3:1 and guarantees that the data transition instruction (DTR) on rung 2:3 will be true, causing the first motion block to be loaded.

Rung 2:3

If the ready bit is high and the sequencer's done bit is low (the module has received a parameter block, but has not yet loaded all the motion blocks), the data transition instruction (DTR) compares the block transfer toggle bit in the status block with the state of the toggle bit from the previous scan. If they differ, the next motion block's address is sequenced into the BTW's control block causing the programmable controller to send the next motion block when rung 2:5 is executed.

Figure 10.15
Program Rungs for Application Program # 2

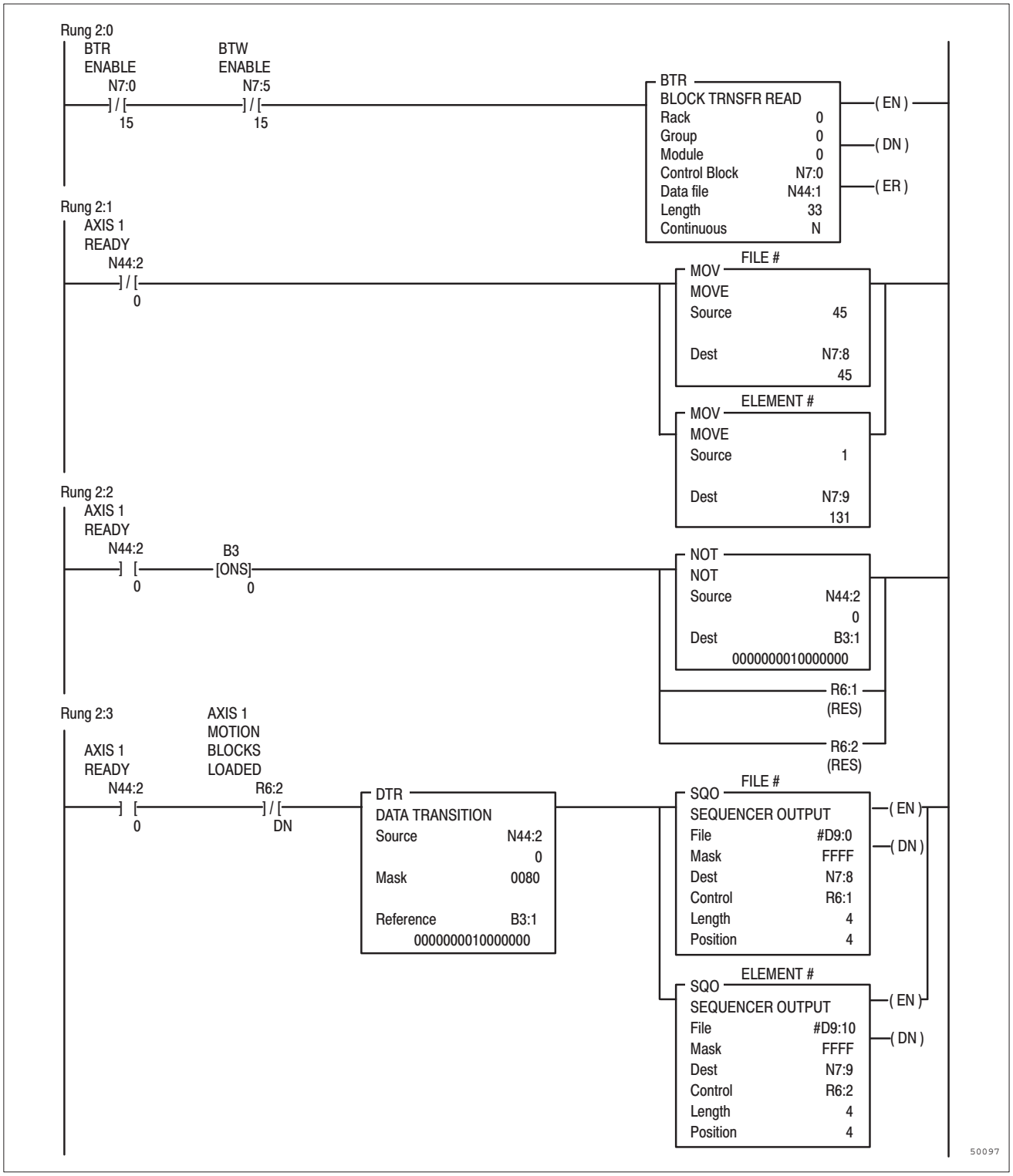
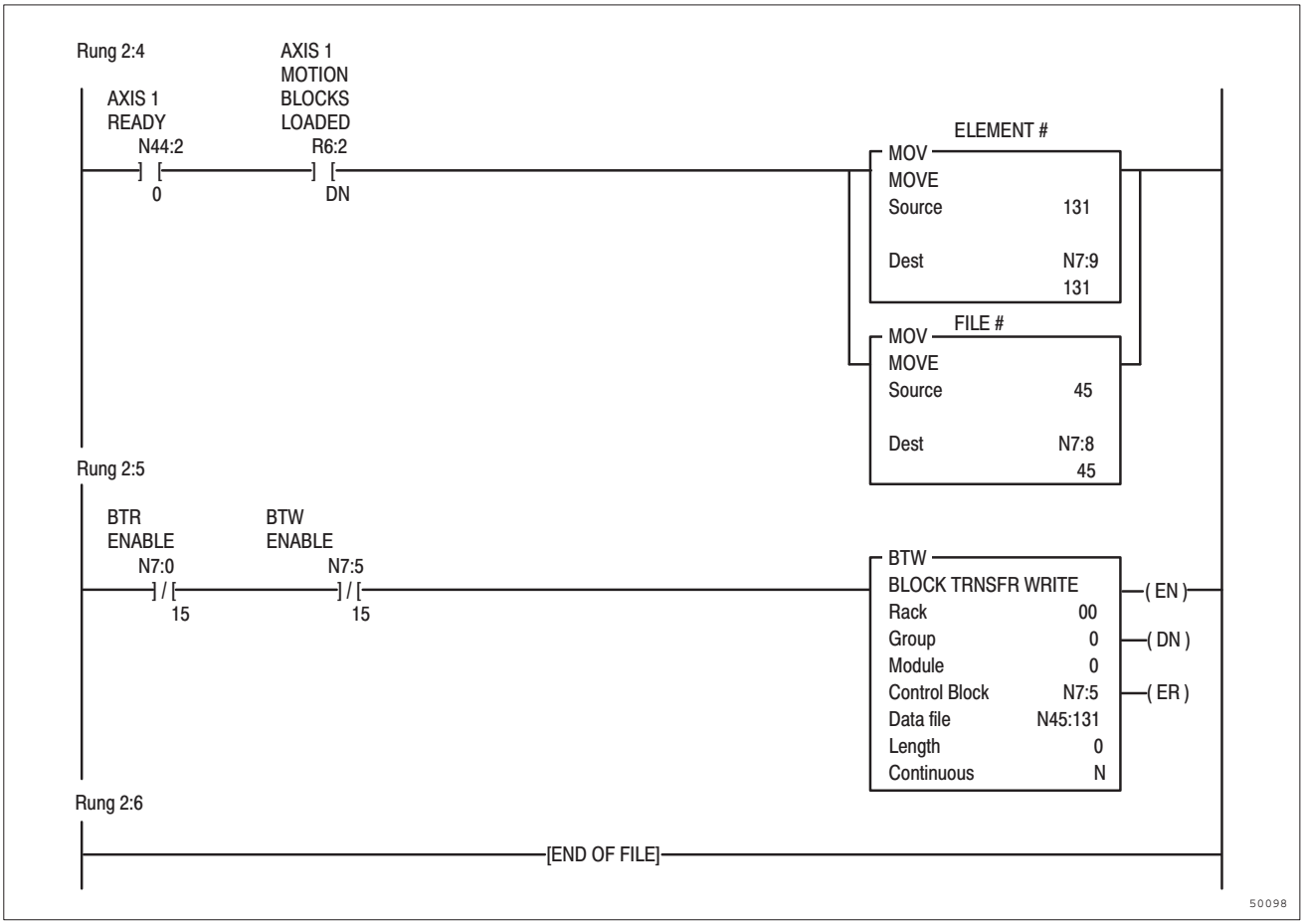


Figure 10.16
Program Rungs for Application Program # 2 (continued)



Rung 2:4

If both the ready bit and the sequencer done bit are high (the module has received a valid parameter block and has loaded all the motion blocks), rung 2:4 moves the source address of the command block into the BTW's control block causing the programmable controller to send a command block when rung 2:5 is executed.

Rung 2:5

Rung 2:5 writes the block selected by rung 2:1, 2:3 or 2:4 to the module.

Chapter 11

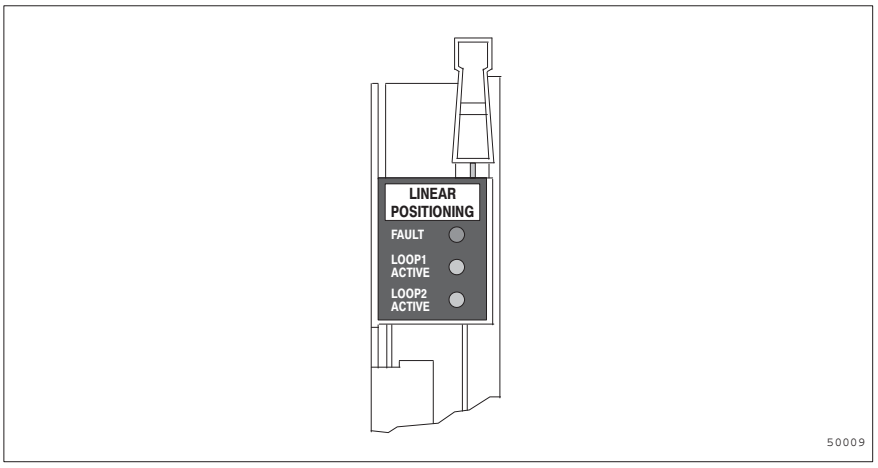
Troubleshooting

The module transfers diagnostic information to the programmable controller in the status block. In addition, the module displays fault information for each loop on the status indicators. Unless the module loses backplane power, all fault conditions cause the fault indicator to turn on. Use the module's indicators and the status block to diagnose and remedy module faults and errors.

Fault Indicators

There are three indicators on the module front panel. (See Figure 11.1.) Monitor them for the occurrence of faults.

Figure 11.1
Fault Indicators



When you power up the module, all three indicators turn on for about one second. Next, the **LOOP 1 ACTIVE** and **LOOP 2 ACTIVE** indicators turn off while the module performs powerup diagnostics. If these diagnostics discover a module fault, the red **FAULT** indicator remains on and the module remains inactive.

After the successful completion of these diagnostics, all indicators go off. The module won't accept any loop parameters until after it detects power at each of the external interfaces. This allows you to turn on the power supplies in any order without generating loop faults. You can, however, still determine which interfaces aren't receiving power by reading loop 1's analog, feedback, and discrete fault bits in status word 2. (See Chapter 6.)

Module Fault Indicator

This red indicator is normally off. It turns on if there is a module fault in one loop or both loops. Faults may be caused by:

- loss of analog power
- analog interface fault
- memory fault
- discrete input fault
- transducer interface fault
- excess following error
- excess PID error
- loss of feedback
- hardware stop input
- immediate stop command

Loop Active Indicators

Each of these green indicators is on when the corresponding loop is active. The indicator blinks if a loop fault occurs and turns off if the loop is inactive.

Loop faults may be caused by loop failure or improper loop configuration.

Indicator Troubleshooting Guide

Refer to Table 11.A when using the indicators to diagnose module problems.

Table 11.A
Troubleshooting Indicators

Indication	Description	Probable Cause	Recommended Action
○ Fault ● Loop 1 ◯ Loop 2	Normal Condition	Module is fully functional.	
○ Fault ○ Loop 1 ○ Loop 2	Power-Up State	A) Powerup complete, awaiting initial parameter block. B) Module not receiving DC power from the chassis backplane or wiring arm terminals.	A) Send parameter block, monitor status block for parameter block errors. B) Check all power supplies, make sure the module is properly seated in the I/O chassis, monitor status block for analog, feedback, or discrete faults.
● Fault ◍ Loop 1 ◯ Loop 2	Loop 1 Fault	A) Analog Interface Fault B) Discrete Input Fault C) Transducer Interface Fault D) Excess Following Error E) Excess PID Error F) Loss of Feedback G) Hardware Stop Input H) Immediate Stop Command	A) Check status block to determine cause of fault. B) Correct fault and issue a reset command to the faulted axes or cycle the I/O chassis power OFF, then ON.
● Fault ◯ Loop 1 ◍ Loop 2	Loop 2 Fault		
● Fault ◍ Loop 1 ◍ Loop 2	Both Loops Faulted		
● Fault ◯ Loop 1 ◯ Loop 2	Module Fault	Internal Circuitry Fault	Cycle power to the I/O chassis containing the module. Return the module for repair if the FAULT LED remains lit when you restore power.
○ Fault ○ Loop 1 ● Loop 2	Loop 1 Inactive Loop 2 Active	Module hasn't received or accepted Loop 1 parameter block.	A) Send Loop 1 parameters. B) Check status block for Loop 1 parameter block errors.
Legend: ● ON ○ OFF ◍ Blinking ◯ ON or OFF			

Troubleshooting Feedback Faults

If you are experiencing feedback faults, perform the following loopback test to determine whether the problem is in the module, the transducer, or the cabling between the module and the transducer:

1. Turn off axis power.
2. Disconnect the transducer from the +GATE (1/2), +INTERR (5/6), -GATE (3/4), and -INTERR (7/8) terminals on the module's wiring arm.
3. Connect the +GATE terminal (1/2) to the +INTERR terminal (5/6).

4. Connect the -GATE terminal (3/4) to the -INTERR terminal (7/8).

5. Power up the axis and check the status block for feedback faults.

If you still experience feedback faults, make sure that your transducer power supply is providing +5 VDC (±5%) through terminals 9 and 10 on the module's wiring arm.

If there aren't any feedback faults, then the problems originated in the transducer or the cabling. Check the cable by performing the loopback test again with the gate and interrogate lines connected at the far end of the cable instead of at the module.

Troubleshooting Flowchart

The flowchart in Figure 11.2 provides a logical procedure to help you isolate a problem with the Linear Positioning Module. You can also use it at system startup to check out the module for proper operation. Numbered explanatory notes follow the flowchart. Many of the flowchart boxes that specify corrective actions contain more than one instruction. When using the flowchart, perform instruction 1 first. If this fails to correct the problem, perform instruction 2, and so on.

Your ladder diagram program should allow you to display parameter, setpoint, motion, command and status blocks on an industrial terminal.

Figure 11.2
Troubleshooting Flowchart

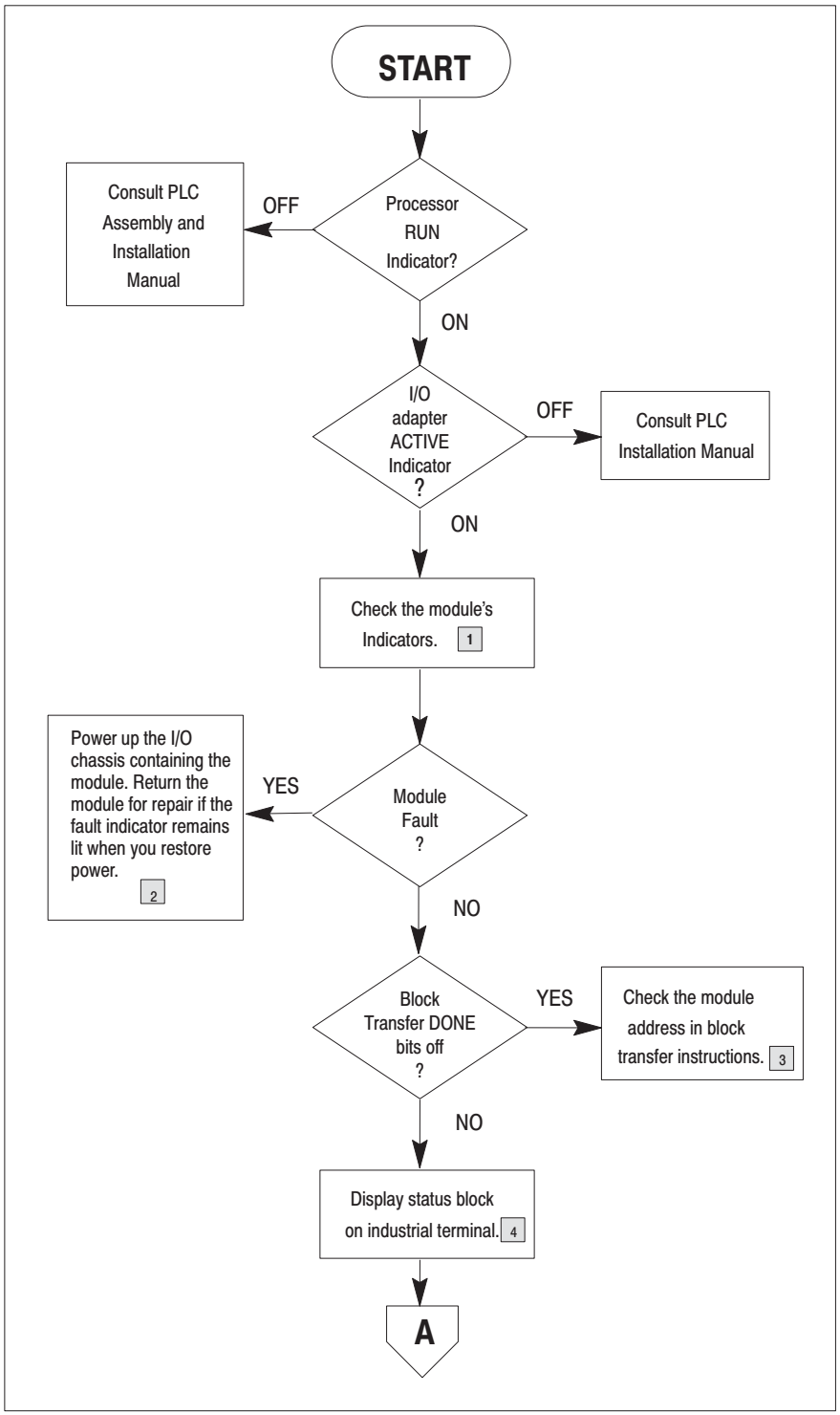

START
Processor RUN Indicator?
OFF
Consult PLC Assembly and Installation Manual
ON
I/O adapter ACTIVE Indicator ?
OFF
Consult PLC Installation Manual
ON
Check the module's Indicators. 1
Module Fault ?
YES
Power up the I/O chassis containing the module. Return the module for repair if the fault indicator remains lit when you restore power. 2
NO
Block Transfer DONE bits off ?
YES
Check the module address in block transfer instructions. 3
NO
Display status block on industrial terminal. 4
A

Figure 11.2
Troubleshooting Flowchart (Continued)

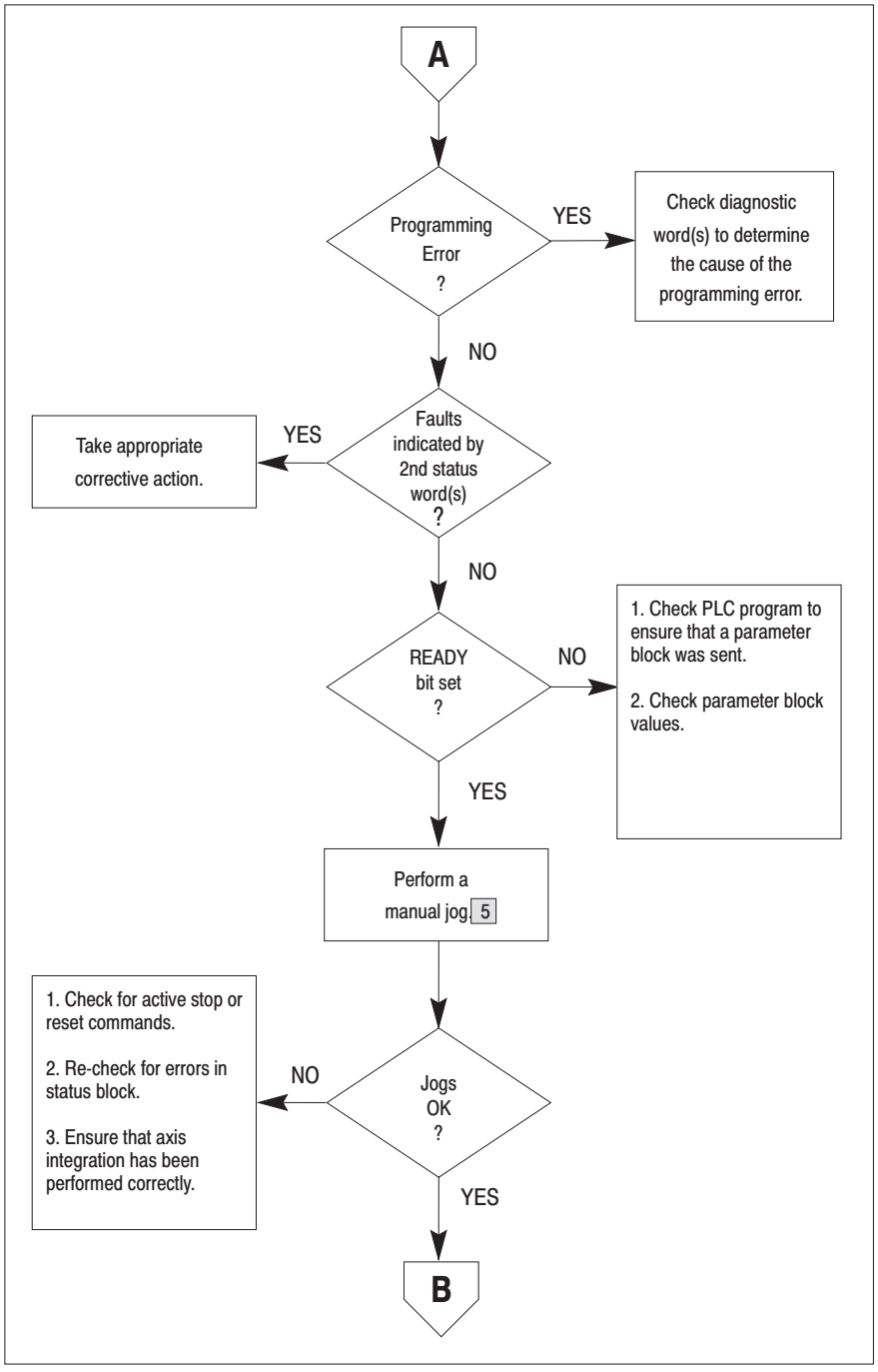
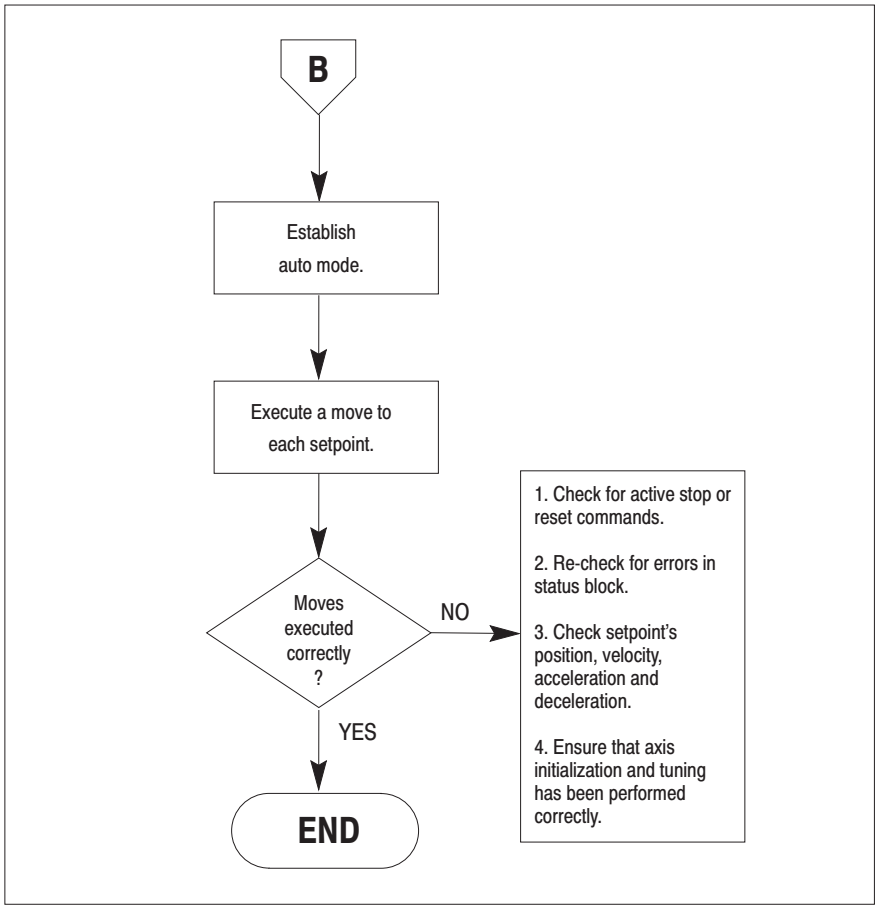

A
Programming Error ?
YES
Check diagnostic word(s) to determine the cause of the programming error.
NO
Faults indicated by 2nd status word(s) ?
YES
Take appropriate corrective action.
NO
READY bit set ?
NO
1. Check PLC program to ensure that a parameter block was sent.
2. Check parameter block values.
YES
Perform a manual jog. 5
Jogs OK ?
NO
1. Check for active stop or reset commands.
2. Re-check for errors in status block.
3. Ensure that axis integration has been performed correctly.
YES
B

Figure 11.2
Troubleshooting Flowchart (Continued)


B
Establish auto mode.
Execute a move to each setpoint.
Moves executed correctly ?
NO
YES
END
1. Check for active stop or reset commands.
2. Re-check for errors in status block.
3. Check setpoint's position, velocity, acceleration and deceleration.
4. Ensure that axis initialization and tuning has been performed correctly.

Flowchart Notes

1. Refer to the Table 11.A.

2. Guard against damage to equipment by powering down the system before removing or installing any module.

3. The module address programmed in the block transfer instruction must be the address of the Linear Positioning Module. The rack number in the module address must match the setting of the I/O chassis switches.

4. Refer to the application programs in Chapter 10 for examples of rungs that perform this function.

5. If you are using hardware jogs, check the status block to ensure that the discrete inputs have been enabled. The status block shows the current state of each discrete input.

Appendix **A**

Glossary of Terms & Abbreviations

Absolute Position: A position described by its distance from the zero point of a coordinate axis.

Acceleration: The rate at which the speed of axis motion increases.

Adapter Module: A module that provides communication between an I/O chassis and the programmable controller. It transmits I/O chassis input data to, and receives output data from, the programmable controller.

Amplifier: A signal gain device whose output is a function of its input.

Analog: Used to describe a physical quantity, such as voltage or position, that normally varies in a continuous manner.

Axis: A principal direction along which the movement of a tool or workpiece occurs.

Axis Movement: The movement of a tool or workpiece along an axis.

Axis Position: The distance along the axis from a pre-defined origin (in the positive or negative direction).

Backplane: A printed circuit board, located in the back of a chassis, that contains a data bus, power bus, and mating connectors for modules installed in the chassis.

BCD: Binary coded decimal.

Binary: A base 2 numbering system.

Binary Coded Decimal: A numbering system used to express individual decimal digits (0 through 9) in four-bit binary notation.

Bit: A binary digit. The smallest unit of information, represented by 1 or 0. A bit is the smallest division of a PLC word.

Block: A set of words handled as a unit.

Block Transfer: A programming technique used to transfer up to 64 data words between an I/O module and the programmable controller.

Circulations: A digital process that involves re-triggering an interrogation pulse a fixed number of times by the return pulse, to provide more counting time for digital counter circuitry, thus improving resolution from a linear displacement transducer system. The on time of the digital interface electronics pulse duration output is multiplied by a specified factor. *Circulation* and *recirculation* are sometimes used interchangeably.

Clear: To erase the contents of a storage device and replace them with zeros.

Closed-Loop: A signal path that compares its output with desired values to regulate system behavior.

Control Loop: A closed-loop that controls the movement of a tool or workpiece along an axis. (The module has two control loops.)

Current Sink: A signal sending device that shunts current to ground.

Current Source: A signal sending device that generates positive current.

D/A: Digital to analog convertor.

Data Table: The part of programmable logic controller memory that contains I/O values and files. The data table is where you monitor, manipulate, and change data to control your system.

Deceleration: The rate at which the speed of axis motion decreases.

Derivative Control: The component that causes an output signal to change as a function of the rate of change of the error signal. Derivative control helps to stabilize the axis movement by opposing changes in positioning error.

Digital: Representation of data in discrete numerical form.

Digital to Analog Conversion: Production of an analog signal whose instantaneous current or voltage is proportional to the value of the digital input.

DIP: Dual in-line package.

Fault: Any malfunction that interferes with normal operation.

Feedback: The signal or data transmitted to the programmable logic controller from a controlled machine to show the machine's response to the command signal.

Feedback Device: An element of a control system (e.g., a transducer) that converts linear motion to an electrical signal for comparison with the command signal.

Feedback Resolution: The smallest increment of dimension that the feedback device can distinguish and reproduce as an electrical output.

Feedback Signal: The measurement signal indicating the value of a directly controlled variable, which is compared to a commanded value to obtain the corrective error signal.

Feedforward Control: Converting information on upstream conditions into corrective commands to minimize the effect of disturbances.

Firmware: A series of instructions in read-only memory. These instructions are for internal processor functions only.

Forward Motion: Axis movement in a positive direction along a coordinate axis.

Gain: The ratio of the output of a system to its input.

Incremental Position: A position described by its distance from the previous position along a coordinate axis.

Initialize: To cause a program or hardware circuit to return to an original state.

Integral Control: The component that causes an output signal to change as a function of the error signal and time duration.

Instability: The state or property of a system where there is an output for which there is no input.

Instruction: A statement that specifies an operation and the values or location of its operands.

Integrator: A device that integrates an input signal, usually with respect to time.

Interface: The boundary between two systems.

I/O: Input/Output

ips/s: Inches per second per second.

ISA: Instruments Society of America

Jog: A control function that provides for the momentary operation of a servo-valve or motor for manual control of axis motion.

Linear Displacement Transducer: A position-sensing transducer mounted along an axis.

Loop: A signal path.

LS: Least significant (word, byte, or bit).

mA: Milliamperes, a unit of measurement for electric current.

Memory: A group of circuit elements that can store data.

Millisecond (ms): One thousandth of a second.

Module: A unit of a larger assembly.

Motion Block: A block containing motion segments and, optionally, a programmable I/O configuration word.

Motion Segment: A movement profile, trigger conditions, and programmable output options that provide an advanced axis motion description.

MS: Most significant (word, byte, or bit).

Noise: An extraneous signal in an electrical circuit capable of interfering with the desired signal.

Open Loop: A signal path without feedback.

Overshoot: The amount that a controlled variable exceeds the desired value after a change of input.

PID: See Proportional, Integral, and Derivative control.

PLC: Programmable Logic Controller. An A-B device that you can program to control and monitor modules in a process control system.

PLC Programming: Storing programs and ladder logic diagrams in the PLC data table.

Proportional Control: The component that causes an output signal to change as a direct ratio of the error signal variation.

RAM: Random access memory.

Read: To acquire data from a source, as in a block transfer of data from an I/O module to the PLC data table.

Read Operation: A PLC request for module status information. This may be in the form of a status block or a status monitor byte.

Register: A memory word or area used for temporary storage of data from an intelligent I/O module to the PLC data table.

Repeatability: The ability to return to the same linear measurement along the same axis.

Reverse Motion: Axis movement in a negative direction along a coordinate axis.

rms: Root mean square.

Servo Valve: A hydraulic valve assembly capable of controlling the linear movement of a tool or workpiece.

Setpoint: A pre-defined position on the axis.

Shield: A conductive barrier that reduces the effect of electric and/or magnetic fields.

Sign: The symbol or bit that distinguishes positive from negative numbers.

Signal: The event or electrical quantity that conveys information from one point to another.

Significant Data: A digit that contributes to the precision of a value. The number of significant digits begins with the one contributing the most value (called the most significant digit) and ends with the digit contributing the least value (called the least significant digit).

Summing Point: A point where signals are added algebraically.

Transducer: A device which receives energy in one form and supplies energy in another; in a positioning system, transducers usually measure linear displacement.

Trigger Conditions: The conditions which, when satisfied, cause a subsequent motion segment to occur.

True: As related to PLC instructions, an enabled logic state.

Velocity: The target speed for the tool or workpiece to move along the axis.

Wiring Arm Terminal: Terminals used to connect to external power supplies, discrete inputs and outputs, and interfaces.

Word: A sequence of bits treated as a unit.

Word Length: The number of bits in a word. The PLC word length is 16 bits.

Write: The process of loading information into memory, as in a block transfer of data from the processor data table to an I/O module.

Write Operation: Sending a parameter block, setpoint block, motion block, or command block from the programmable controller to the module.

Appendix **B**

Status Block

Figure B.1
Status Block Word Assignments

WORD		DESCRIPTION	
AXIS 1	AXIS 2		
1		Module Configuration Word	Default Status
2	(6)	Status word 1	
3	(7)	Status word 2	
4	(8)	(MS) Position/Error/Diagnostic word	
5	(9)	(LS) Position/Error/Diagnostic word	
10	(11)	Active motion segment/setpoint	Extended Status
12	(14)	(MS) Position	
13	(15)	(LS) Position	
16	(18)	(MS) Following Error	
17	(19)	(LS) Following Error	
20	(21)	Measured Velocity	
22	(23)	Desired Velocity	
24	(25)	Desired Acceleration	
26	(27)	Desired Deceleration	
28	(29)	% Analog Output	
30	(31)	Maximum Positive Velocity	
32	(33)	Maximum Negative Velocity	

50000

Figure B.2
Module Configuration Word (word 1)

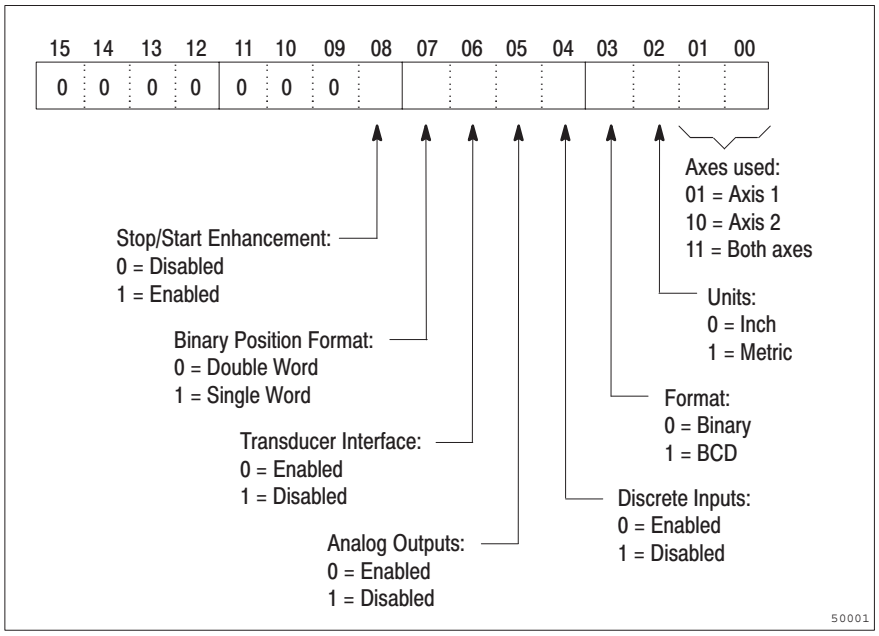


Figure B.3
Status Word 1 (words 2 and 6)

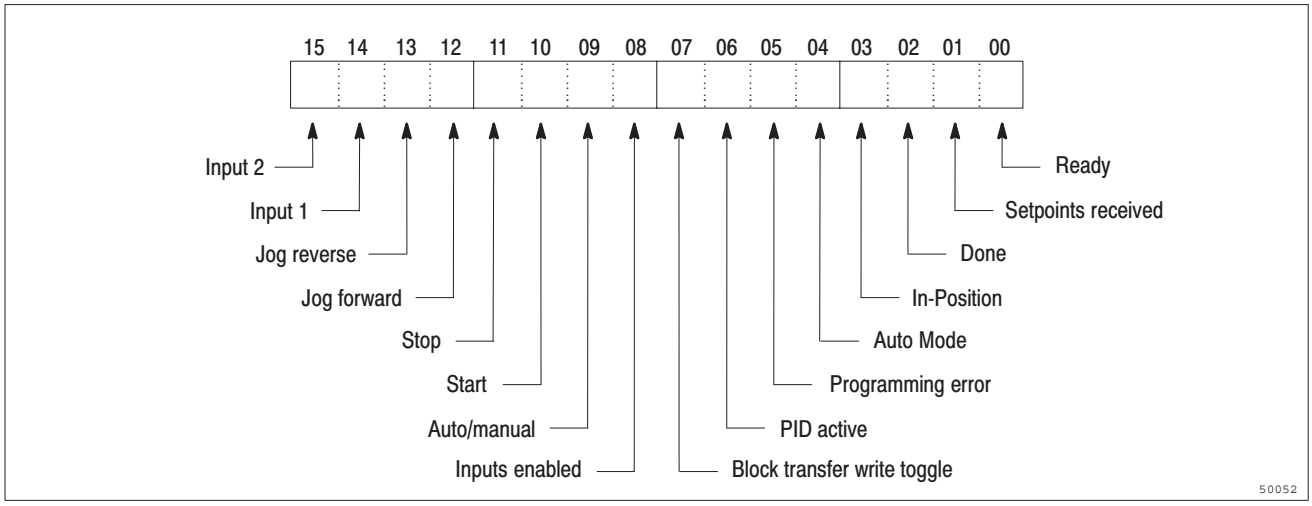


Figure B.4
Status Word 2 (words 3 and 7)

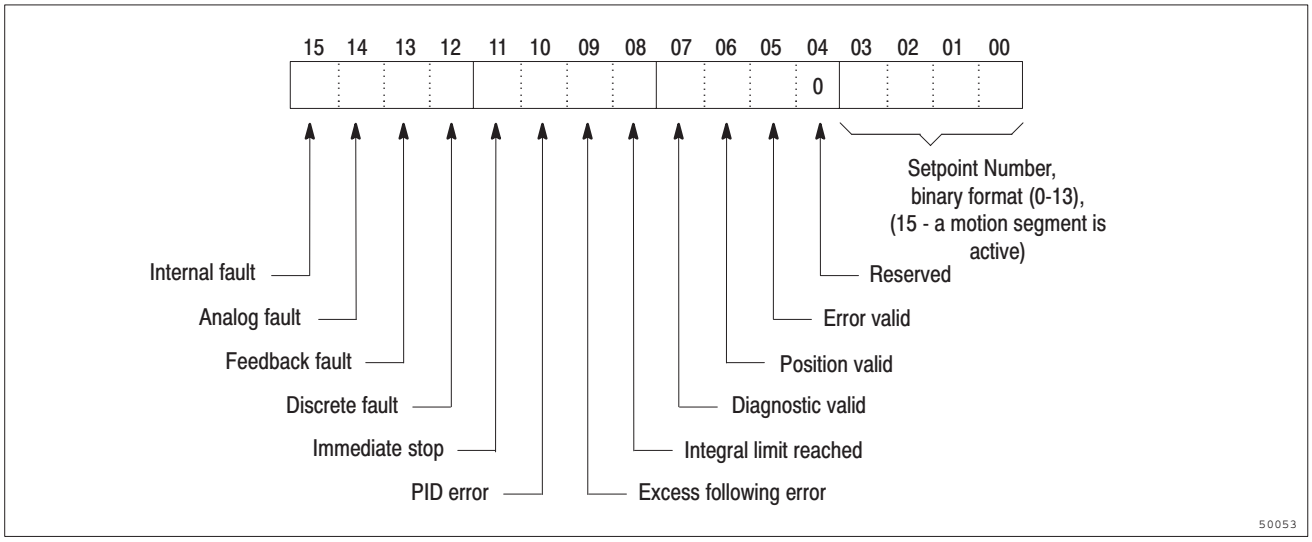


Figure B.5
Position/Error/Diagnostic Words (words 4, 5 and 8, 9)
Diagnostic Format

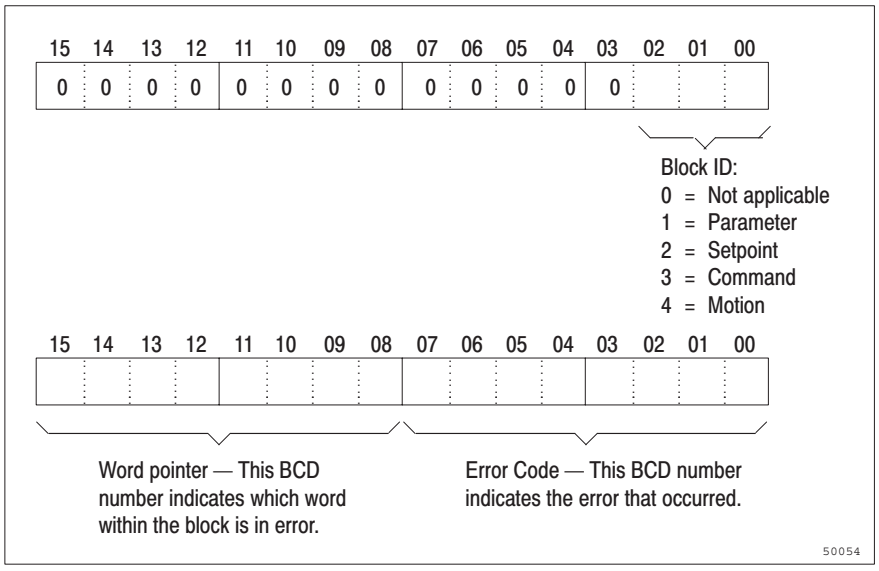

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0 0 0 0 0 0
Block ID:
0 = Not applicable
1 = Parameter
2 = Setpoint
3 = Command
4 = Motion
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Word pointer — This BCD number indicates which word within the block is in error.
Error Code — This BCD number indicates the error that occurred.
50054

Figure B.6
Position/Error/Diagnostic Words (words 4, 5; 8, 9; 12, 13; and 14, 15)
Position Format

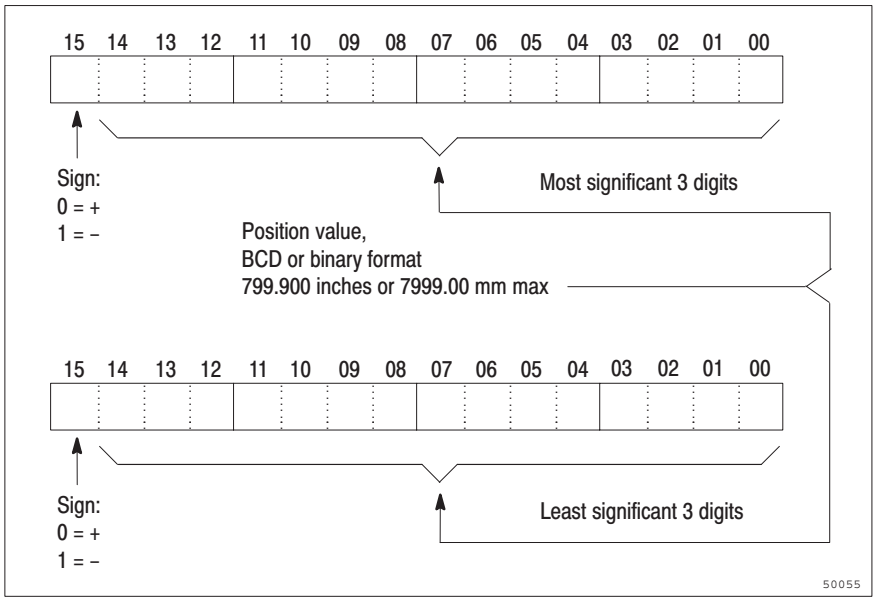

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant 3 digits
Position value,
BCD or binary format
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50055

Figure B.7
Position/Error/Diagnostic Words (words 4, 5; 8, 9; 16,17; and 18, 19)
Following Error Format

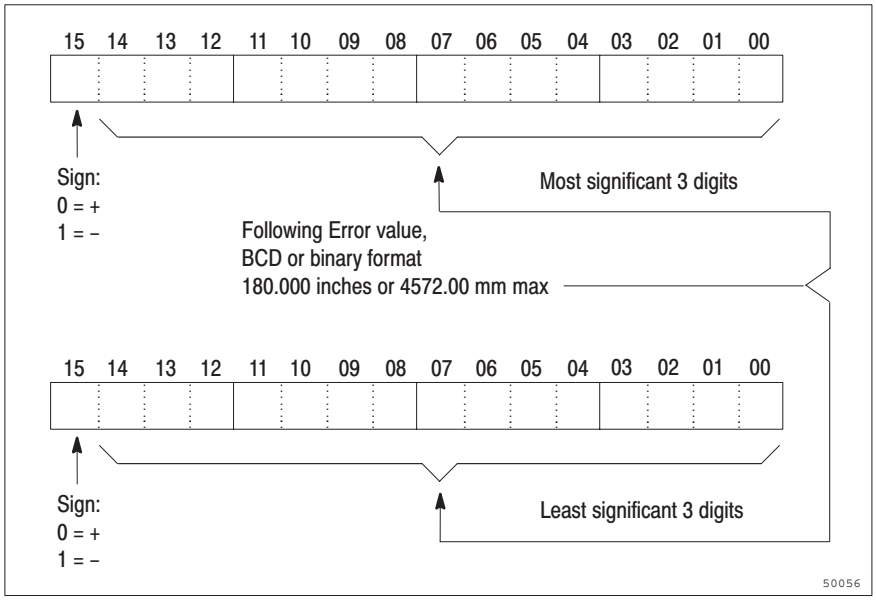

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant 3 digits
Following Error value,
BCD or binary format
180.000 inches or 4572.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50056

Figure B.8
Active Motion Segment/Setpoint (words 10 and 11)

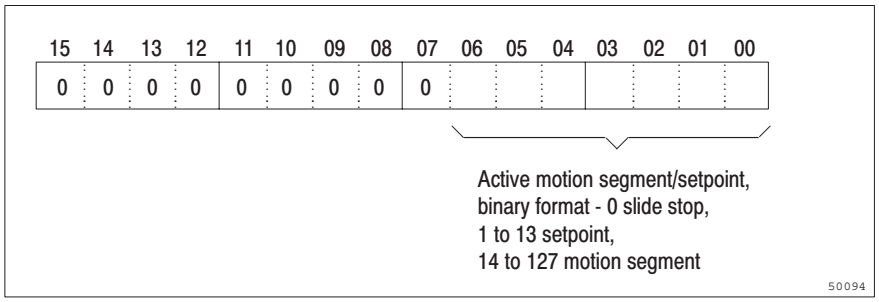

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0 0
Active motion segment/setpoint,
binary format - 0 slide stop,
1 to 13 setpoint,
14 to 127 motion segment
50094

Figure B.9
Measured Velocity (words 20 and 21)

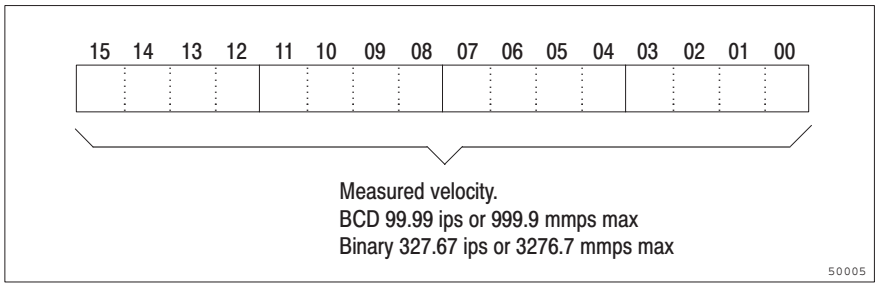


Figure B.10
Desired Velocity (words 22 and 23)

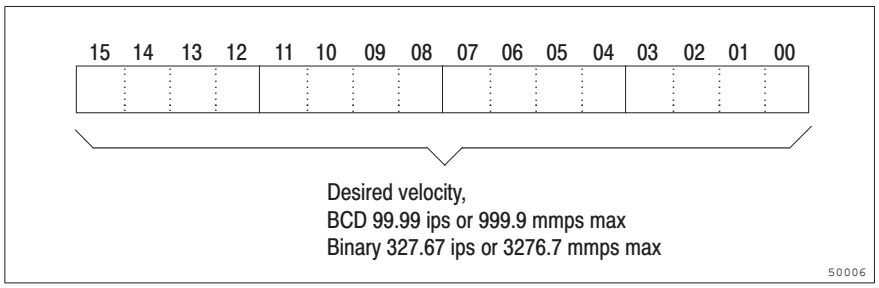


Figure B.11
Desired Acceleration (words 24 and 25)

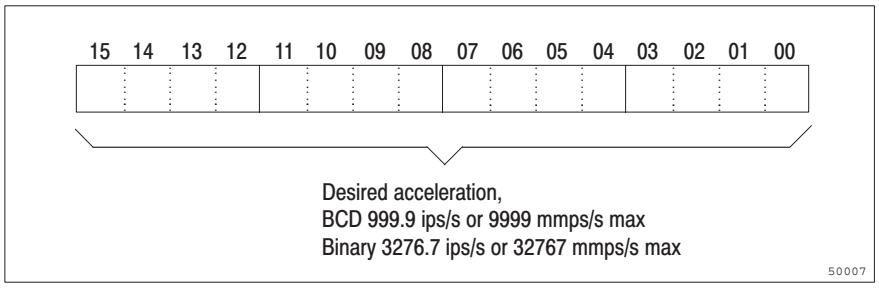

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Desired acceleration,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50007

Figure B.12
Desired Deceleration (words 26 and 27)

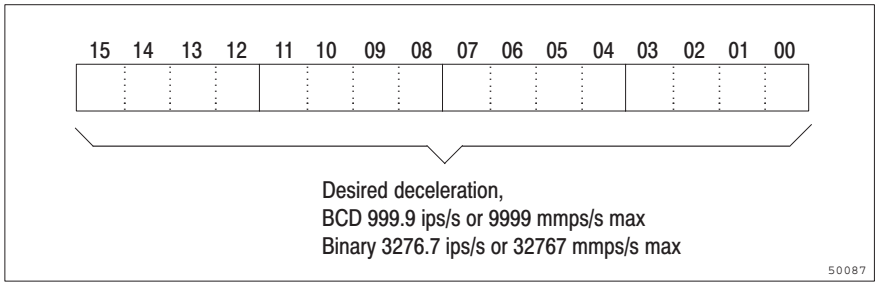

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Desired deceleration,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50087

Figure B.13
Percent Analog Output (words 28 and 29)

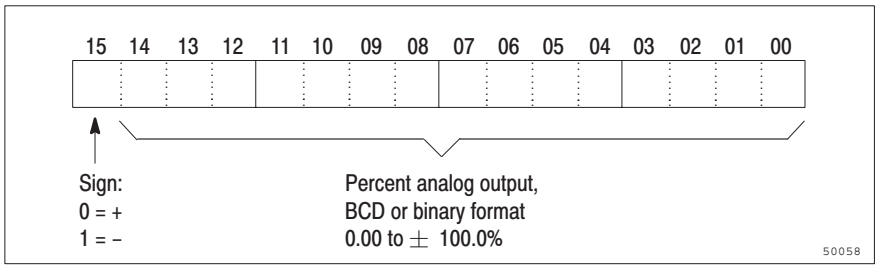


Figure B.14
Maximum Velocity (words 30, 31 and 32, 33)

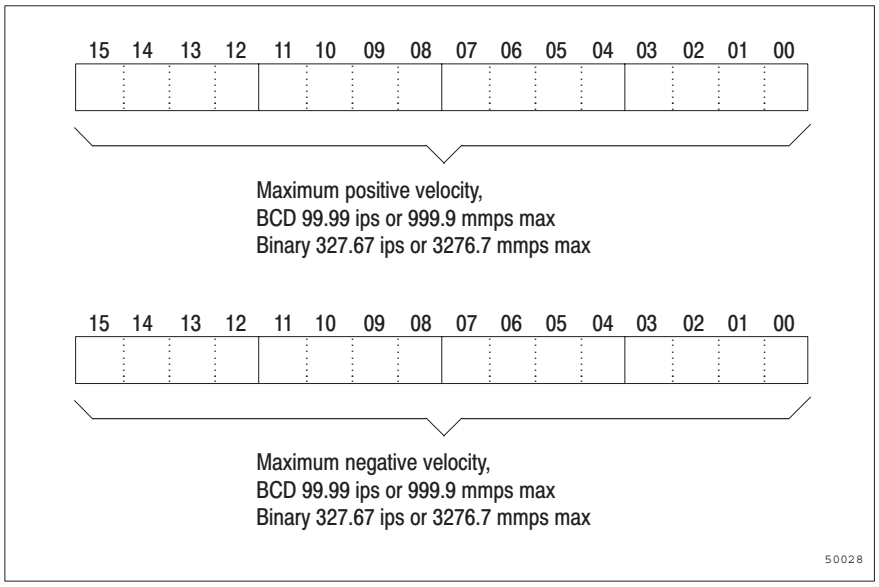

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum positive velocity,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum negative velocity,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
50028

Table B.A
Error Codes

Code	Definition
00	No errors detected
01	Invalid block identifier
02	Non-BCD number entered
03	Invalid bit setting, unused bits must be set to zero
04	Data is out of range
05	Invalid number of axes programmed
06	Setpoint is not defined
07	Setpoint commanded while in manual mode
08	Position exceeds a software overtravel limit
09	Attempted to switch to auto mode with axis in motion
10	Attempted to switch to manual mode with axis in motion
11	Velocity exceeds maximum
12	High jog rate > maximum velocity
13	Low jog rate > high jog rate
14	Maximum PID error must be outside the PID band
15	Incorrect block length
16	First block after powerup must be a parameter block
17	Negative travel limit $\geq$ positive travel limit
18	Jog commanded while in auto mode
19	Forward and reverse jogs commanded simultaneously
20	Block transfer write attempted before module confirmed all power on wiring arm
21	Specified velocity exceeds maximum velocity for direction of motion
22	Motion segment ID not defined
23	Motion segment commanded while in manual mode
24	A motion segment is attempting to use an output which is not configured as a programmable output
25	Motion segment ID previously defined in same motion block

Appendix **C**

Parameter Block

Figure C.1
Parameter Block Word Assignments

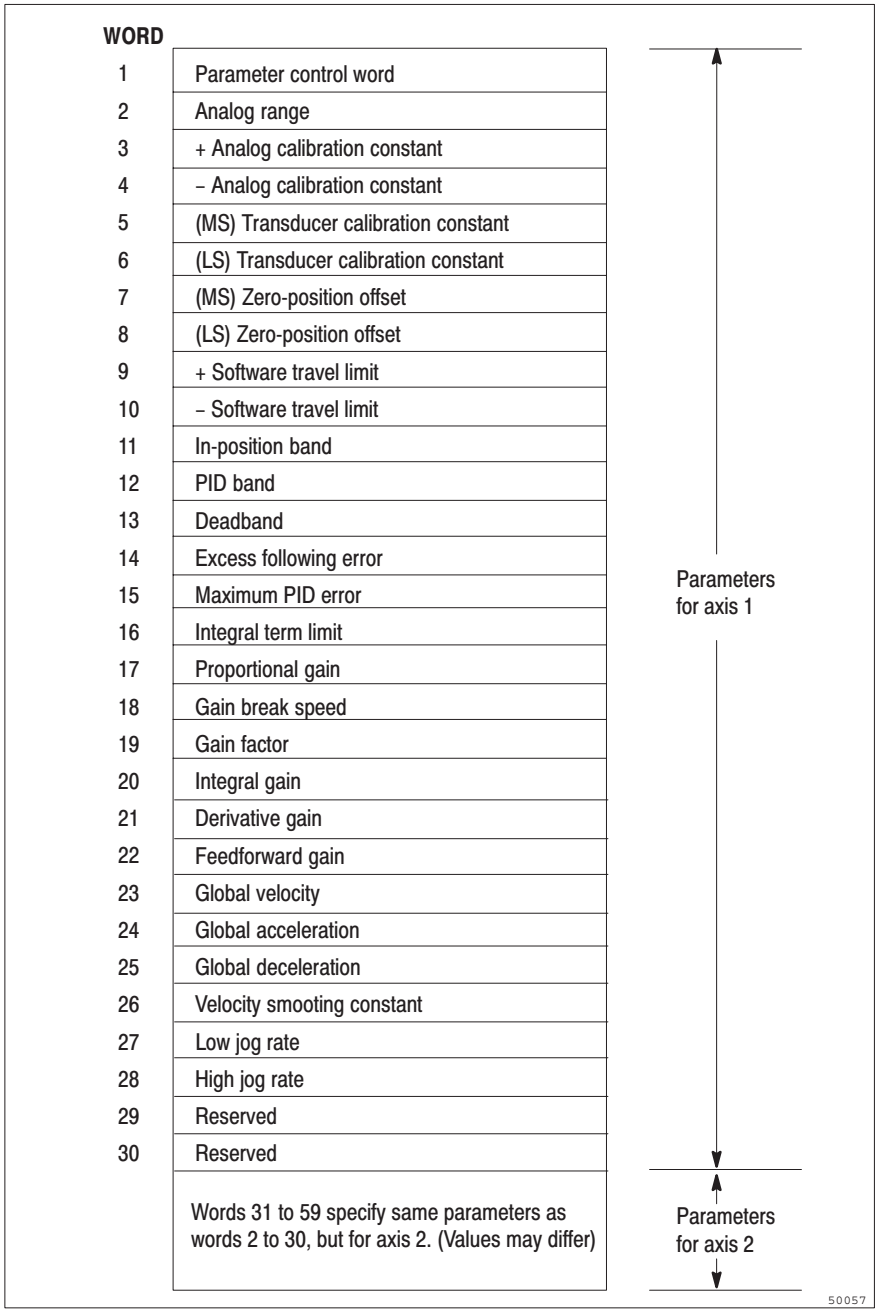


Figure C.2
Parameter Block Control Word (word 1)

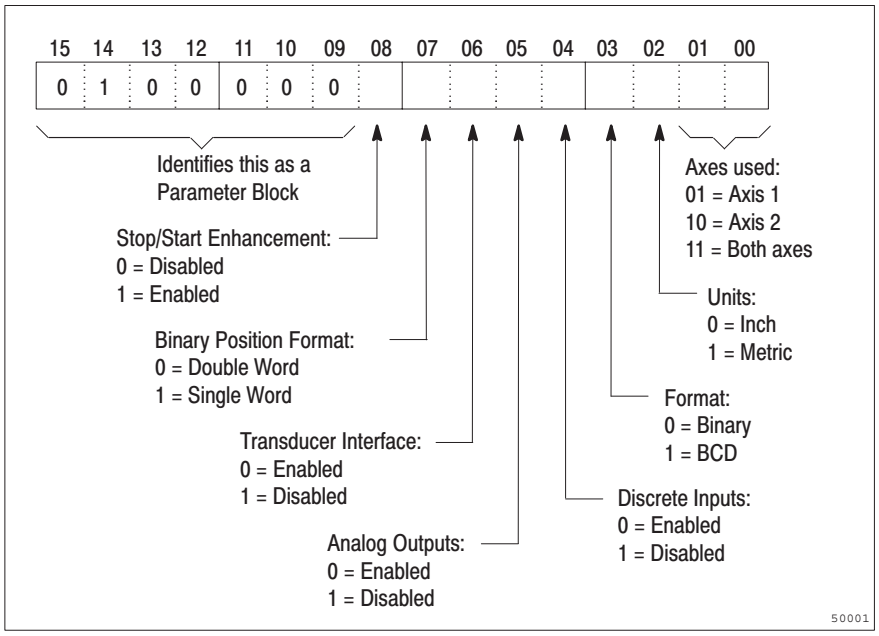


Figure C.3
Analog Range Word (words 2 and 31)

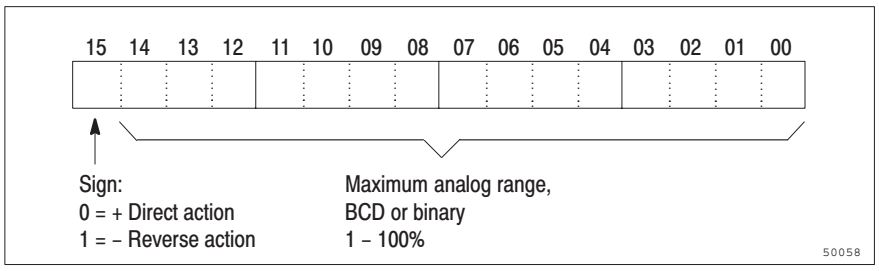

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = + Direct action
1 = – Reverse action
Maximum analog range,
BCD or binary
1 – 100%
50058

Figure C.4
Analog Calibration Constant Words (words 3, 4 and 32, 33)

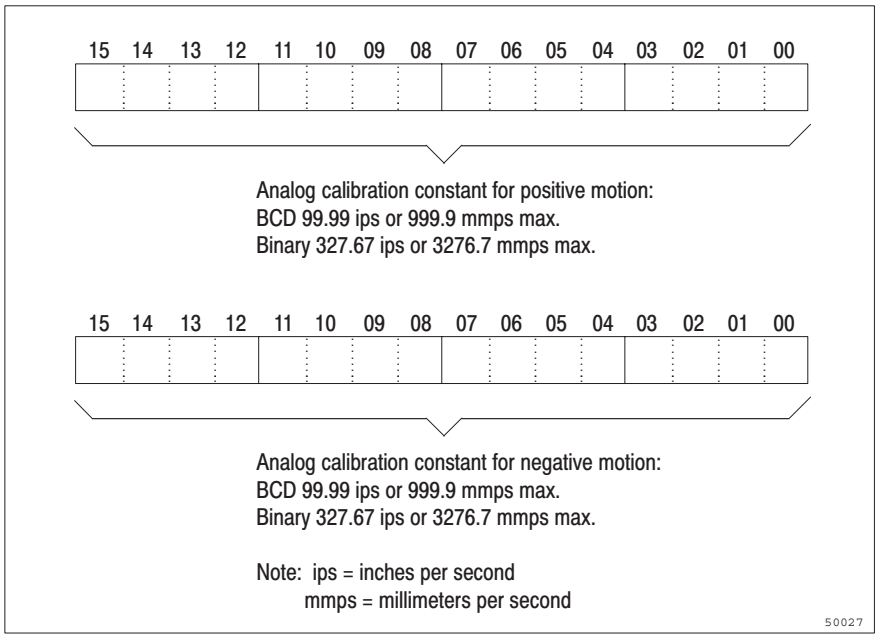

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Analog calibration constant for positive motion:
BCD 99.99 ips or 999.9 mmps max.
Binary 327.67 ips or 3276.7 mmps max.
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Analog calibration constant for negative motion:
BCD 99.99 ips or 999.9 mmps max.
Binary 327.67 ips or 3276.7 mmps max.
Note: ips = inches per second
mmps = millimeters per second
50027

Figure C.5
Transducer Calibration Constant Words (words 5, 6 and 34, 35)

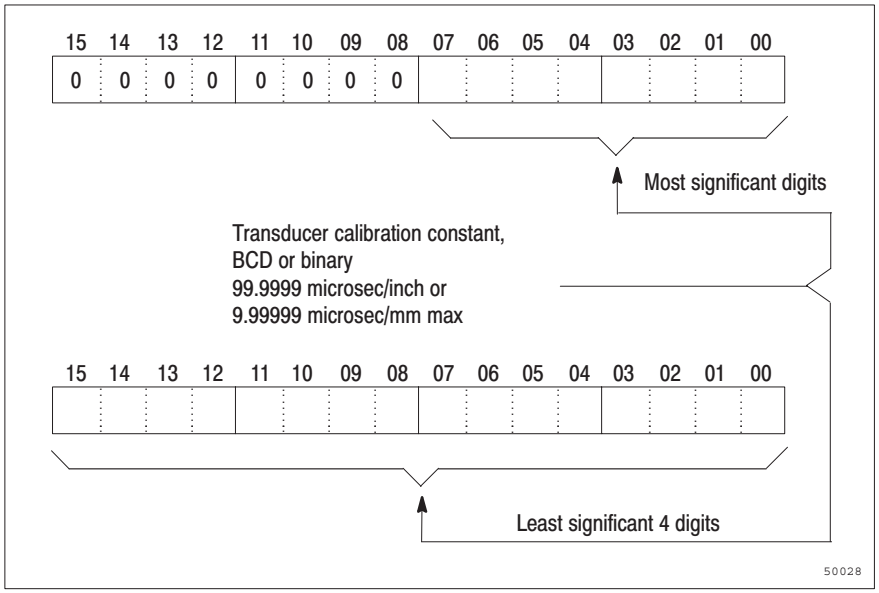

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 0 0 0 0 0 0
Most significant digits
Transducer calibration constant,
BCD or binary
99.9999 microsec/inch or
9.99999 microsec/mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Least significant 4 digits
50028

Figure C.6
Zero-Position Offset Words (words 7, 8 and 36, 37)

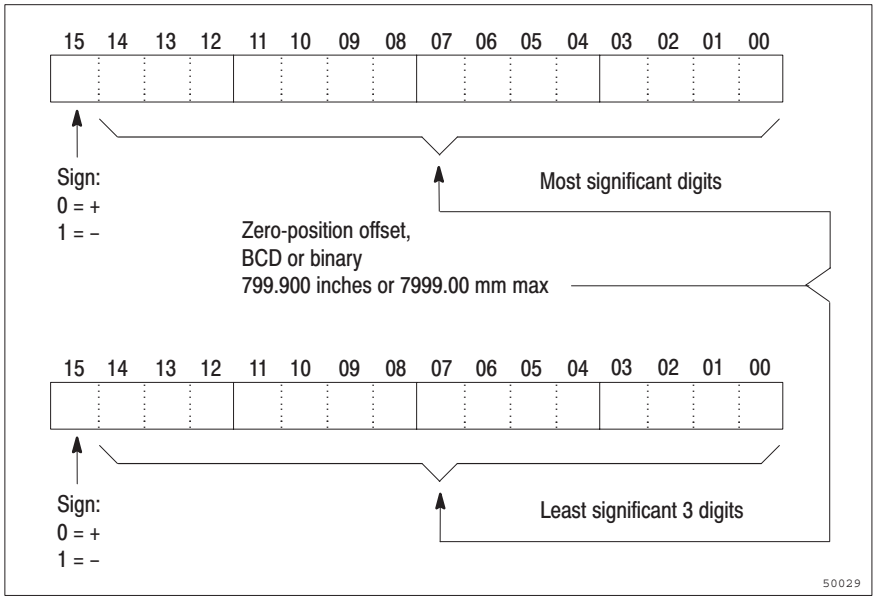

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Zero-position offset,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50029

Figure C.7
Software Travel Limit Words (words 9, 10 and 38, 39)

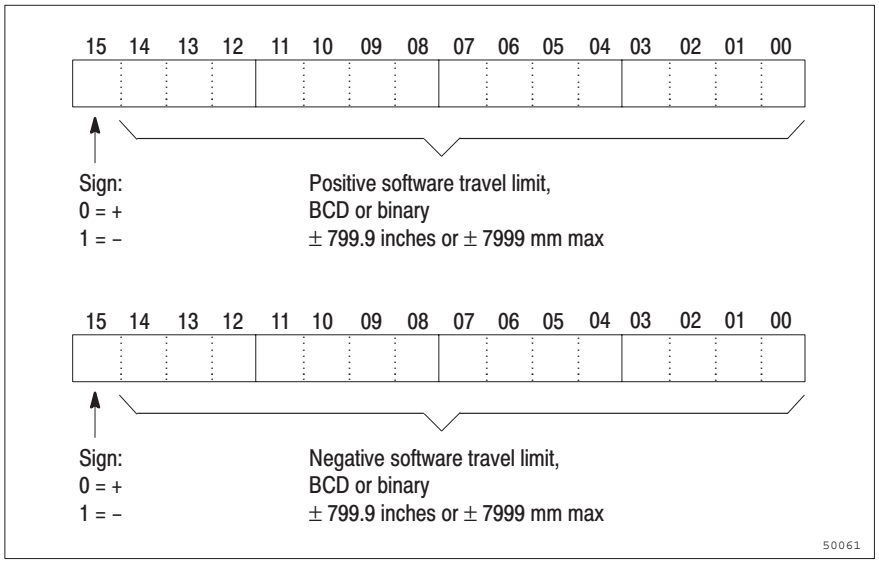

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Positive software travel limit,
BCD or binary
± 799.9 inches or ± 7999 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Negative software travel limit,
BCD or binary
± 799.9 inches or ± 7999 mm max
50061

Figure C.8
In-Position Band Word (words 11 and 40)

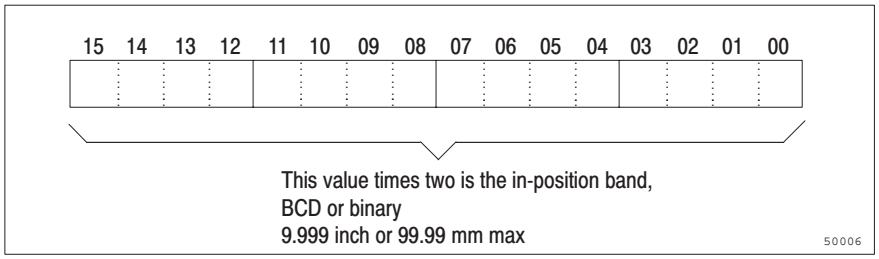

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
This value times two is the in-position band,
BCD or binary
9.999 inch or 99.99 mm max
50006

Figure C.9
PID Band Word (words 12 and 41)

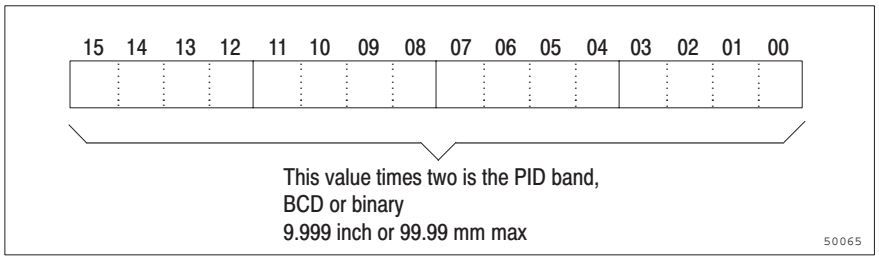


Figure C.10
Deadband Word (words 13 and 42)

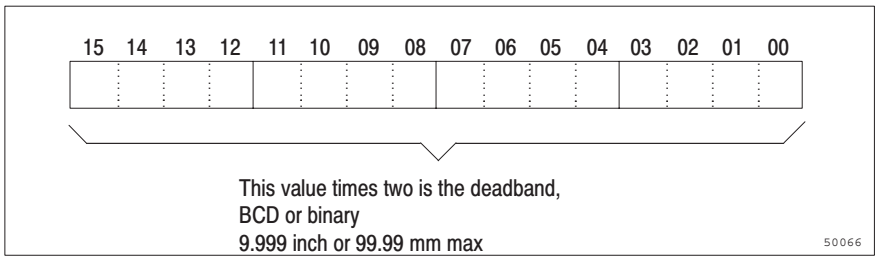


Figure C.11
Excess Following Error Word (words 14 and 43)

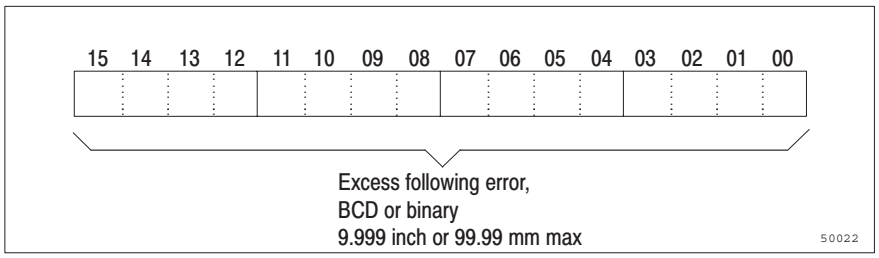


Figure C.12
Maximum PID Error Word (words 15 and 44)

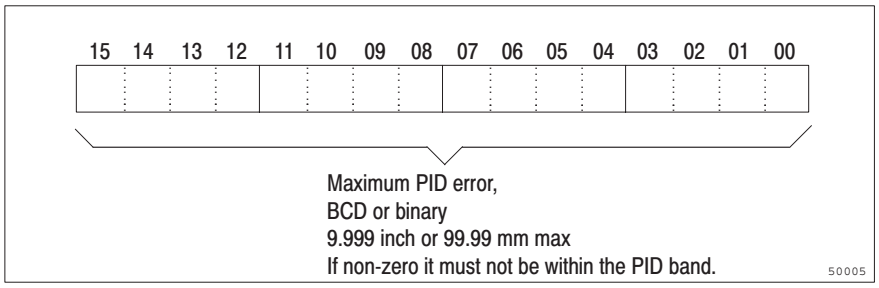

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Maximum PID error,
BCD or binary
9.999 inch or 99.99 mm max
If non-zero it must not be within the PID band.
50005

Figure C.13
Integral Term Limit Word (words 16 and 45)

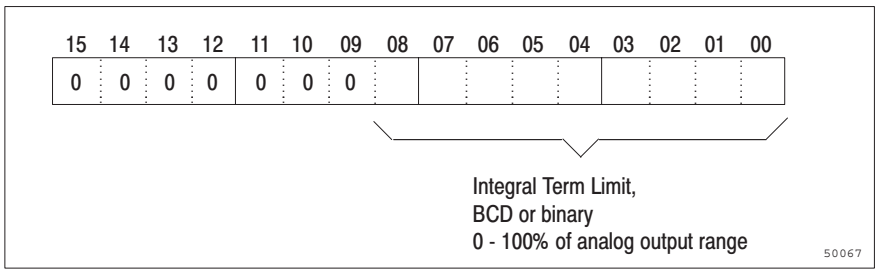


Figure C.14
Proportional Gain Word (words 17 and 46)

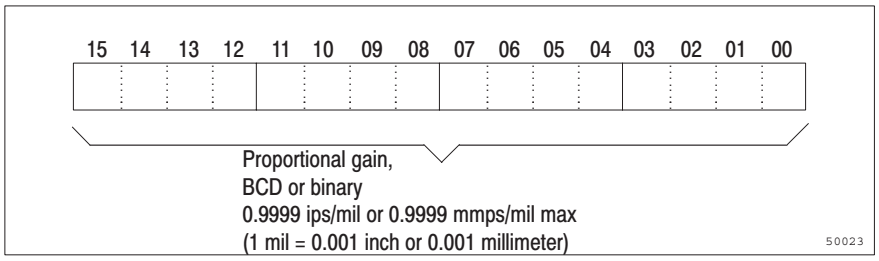

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Proportional gain,
BCD or binary
0.9999 ips/mil or 0.9999 mmps/mil max
(1 mil = 0.001 inch or 0.001 millimeter)
50023

Figure C.15
Gain Break Speed Word (words 18 and 47)

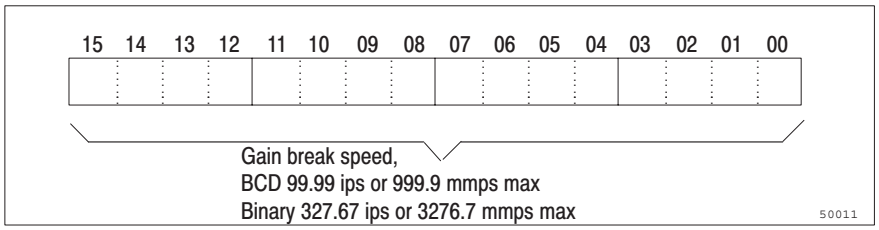


Figure C.16
Gain Factor Word (words 19 and 48)

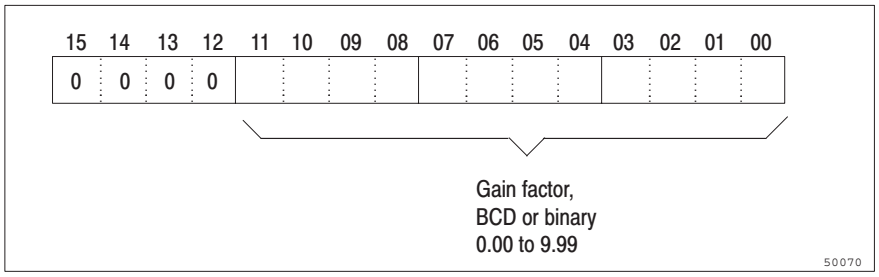


Figure C.17
Integral Gain Word (words 20 and 49)

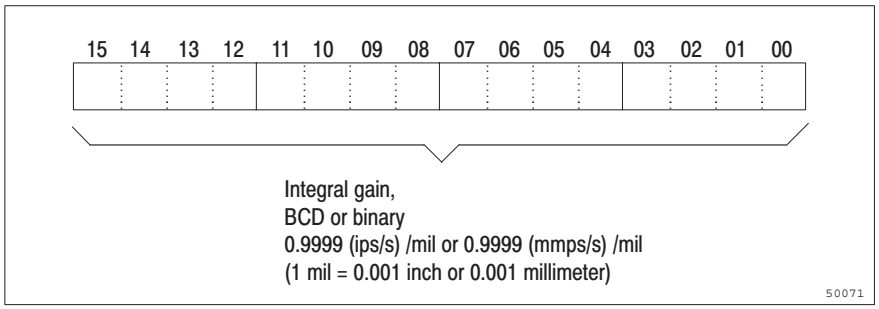

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Integral gain,
BCD or binary
0.9999 (ips/s) /mil or 0.9999 (mmps/s) /mil
(1 mil = 0.001 inch or 0.001 millimeter)
50071

Figure C.18
Derivative Gain Word (words 21 and 50)

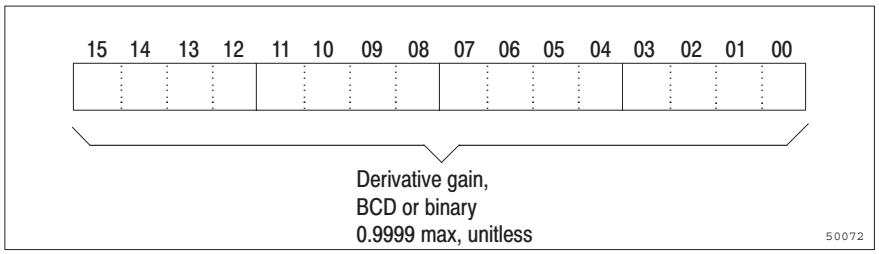


Figure C.19
Feedforward Gain Word (words 22 and 51)

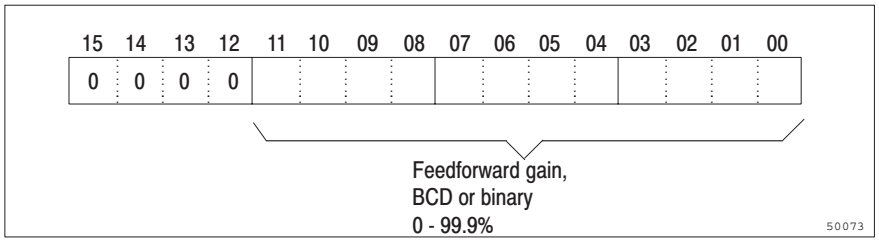


Figure C.20
Global Velocity Word (words 23 and 52)

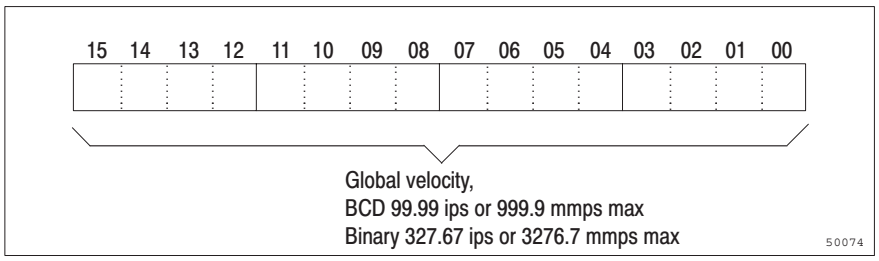


Figure C.21
Global Acceleration/Deceleration Words (words 24, 25 and 53, 54)

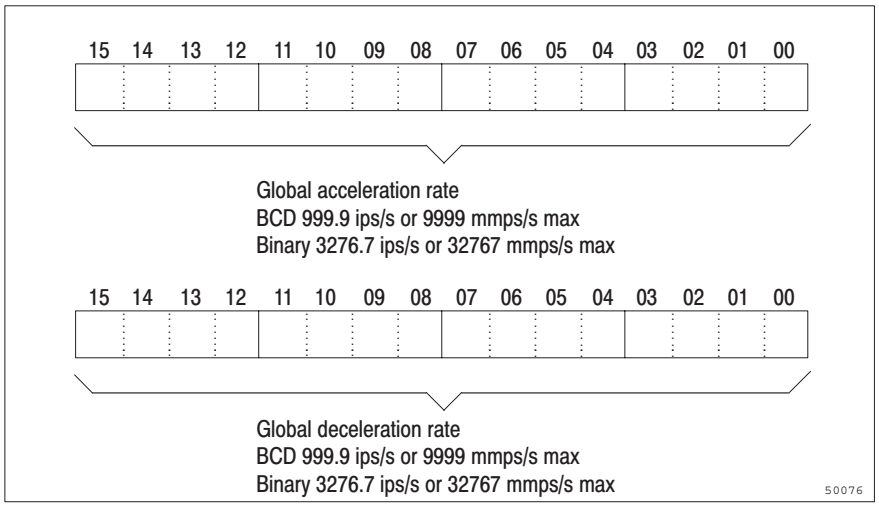

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Global acceleration rate
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Global deceleration rate
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50076

Figure C.22
Velocity Smoothing (Jerk) Constant Word (words 26 and 55)

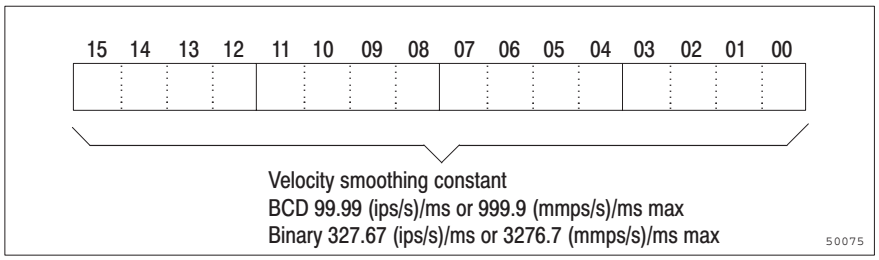

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Velocity smoothing constant
BCD 99.99 (ips/s)/ms or 999.9 (mmps/s)/ms max
Binary 327.67 (ips/s)/ms or 3276.7 (mmps/s)/ms max
50075

Figure C.23
Jog Rate (Low and High) Words (words 27, 28 and 56, 57)

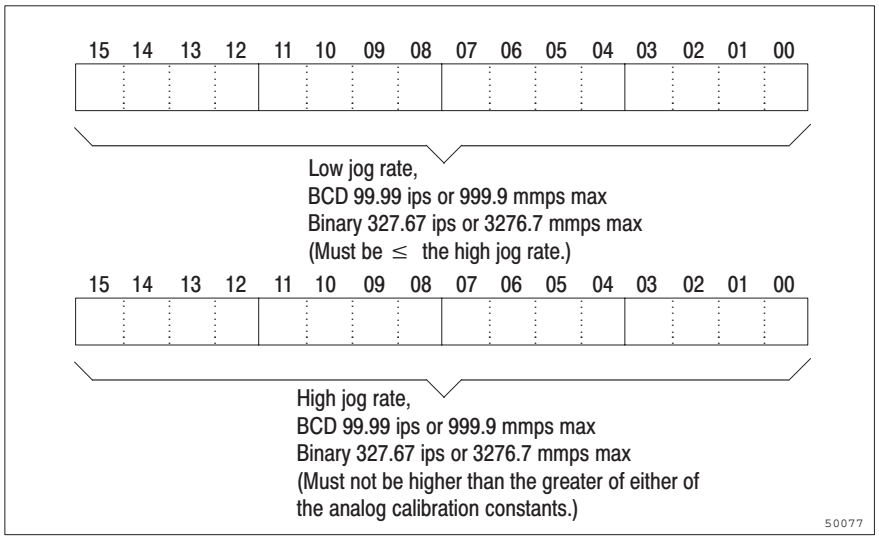

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Low jog rate,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
(Must be ≤ the high jog rate.)
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
High jog rate,
BCD 99.99 ips or 999.9 mmps max
Binary 327.67 ips or 3276.7 mmps max
(Must not be higher than the greater of either of
the analog calibration constants.)
50077

Table C.A
Parameter Block Values

Parameter	Limits	
Analog Range	±1% to ±100%	
+ Analog Calibration Constant	0 to 327.67 ips	0 to 3276.7 mmps
– Analog Calibration Constant	0 to 327.67 ips	0 to 3276.7 mmps
Transducer Calibration Constant	0.0001 to 99.9999 microsec./in.	0.00001 to 9.99999 microsec./mm
Zero-Position Offset	–799.900 to +799.900 in.	–7999.00 to +7999.00 mm
+ Software Travel Limit	–799.9 to +799.9 in.	–7999 to +7999 mm
– Software Travel Limit	–799.9 to +799.9 in.	–7999 to +7999 mm
In-Position Band	0 to 9.999 in.	0 to 99.99 mm
PID Band	0 to 9.999 in.	0 to 99.99 mm
Deadband	0 to 9.999 in.	0 to 99.99 mm
Excess Following Error	0 to 9.999 in.	0 to 99.99 mm
Maximum PID Error	0 to 9.999 in.	0 to 99.99 mm
Integral Term Limit	0 to 100%	
Proportional Gain	0 to 0.9999 ips/mil	0 to 0.9999 mmps/mil
Gain Break Speed	0 to 327.67 ips	0 to 3276.7 mmps
Gain Factor	0.00 to 9.99	
Integral Gain	0 to 0.9999 (ips/s)/mil	0 to 0.9999 (mmps/s)/mil
Derivative Gain	0 to 0.9999	
Feedforward Gain	0 to 99.9%	
Global Velocity	0 to 327.67 ips	0 to 3276.7 mmps
Global Acceleration	0 to 3276.7 ips/s	0 to 32767 mmps/s
Global Deceleration	0 to 3276.7 ips/s	0 to 32767 mmps/s
Velocity Smoothing Constant	0 to 327.67 (ips/s)/ms	0 to 3276.7 (mmps/s)/ms
Low Jog Rate	0 to 327.67 ips	0 to 3276.7 mmps
High Jog Rate	0 to 327.67 ips	0 to 3276.7 mmps
Reserved	Set to Zero	
Reserved	Set to Zero	

Appendix D

Setpoint Block

Figure D.1
Setpoint Block Word Assignments

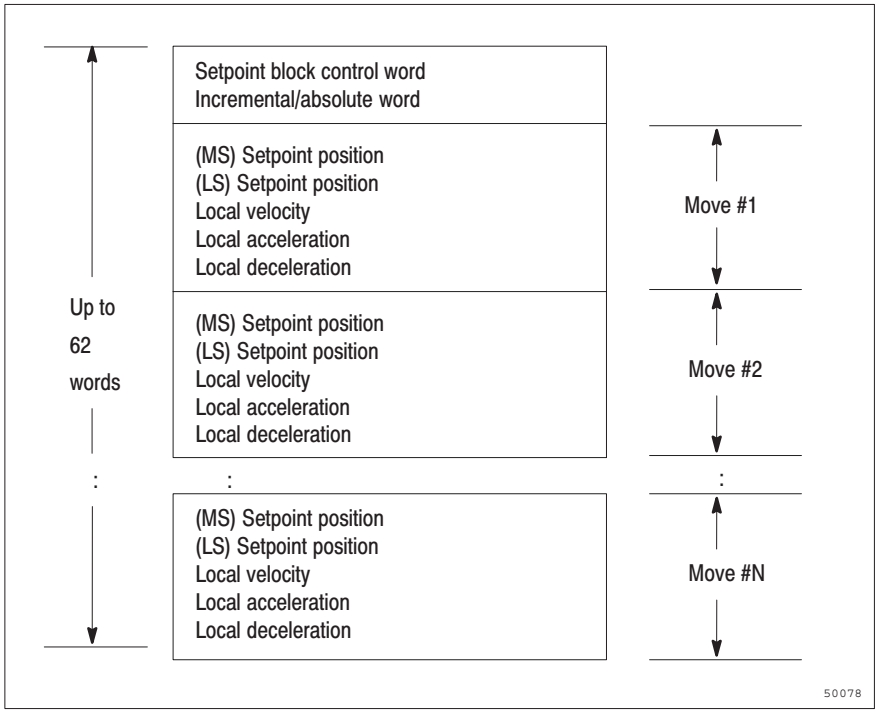


Figure D.2
Setpoint Block Control Word (word 1)

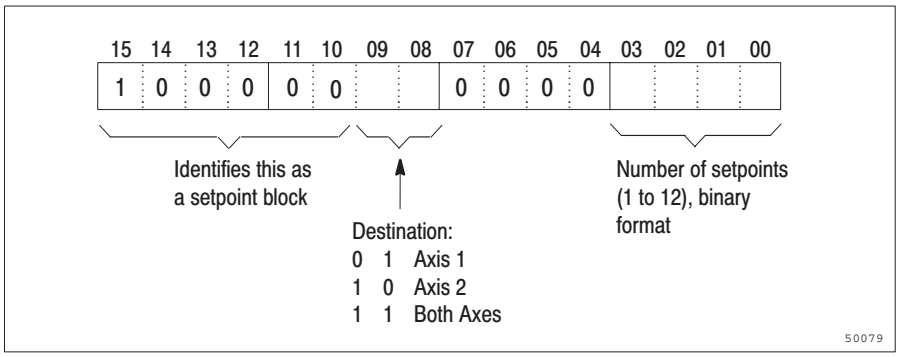

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
1 0 0 0 0 0 0 0 0 0
Identifies this as
a setpoint block
Destination:
0 1 Axis 1
1 0 Axis 2
1 1 Both Axes
Number of setpoints
(1 to 12), binary
format
50079

Figure D.3
Incremental/Absolute Word (word 2)

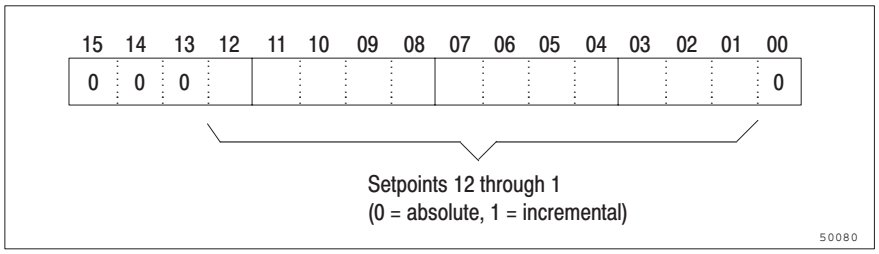


Figure D.4
Setpoint Position Words

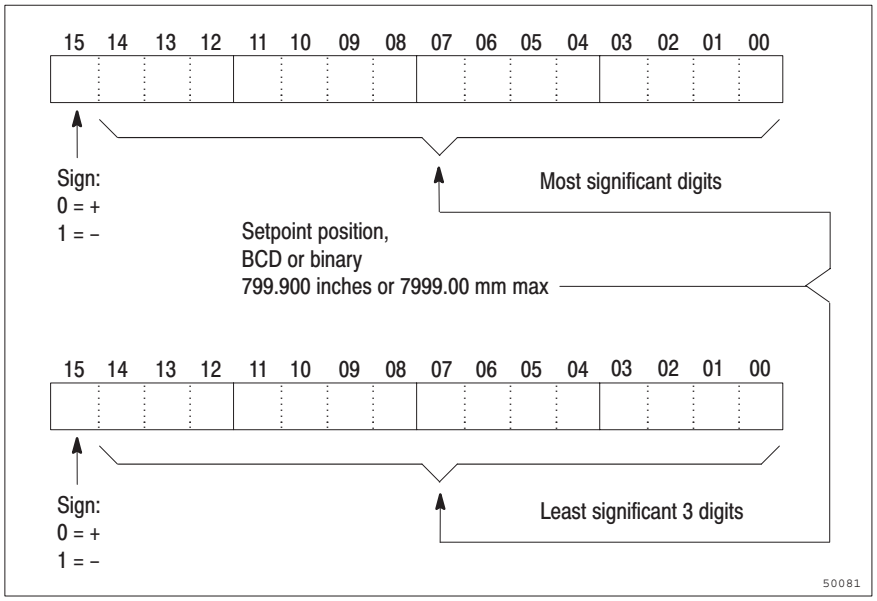

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Setpoint position,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50081

Figure D.5
Local Velocity Words

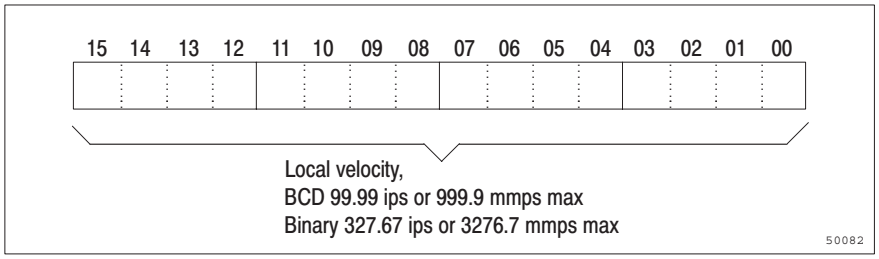
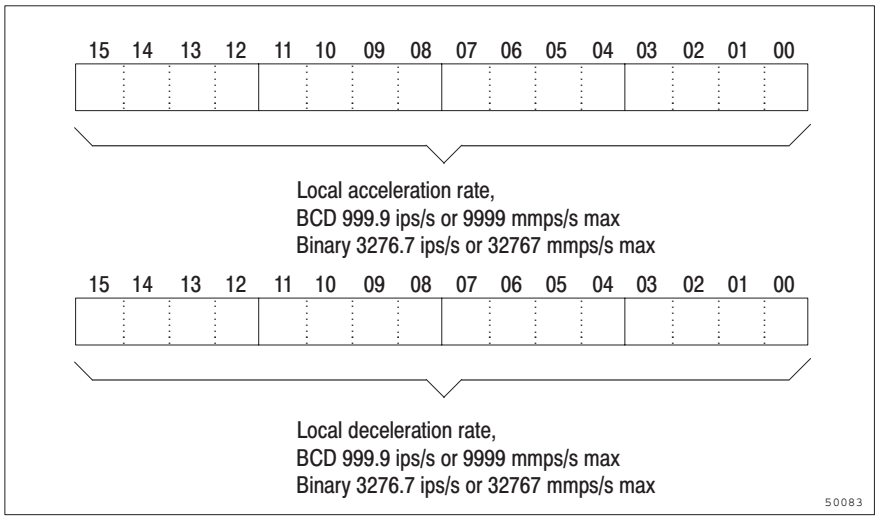


Figure D.6
Local Acceleration/Deceleration Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local acceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local deceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50083

Appendix E

Command Block

Figure E.1
Command Block Word Assignments

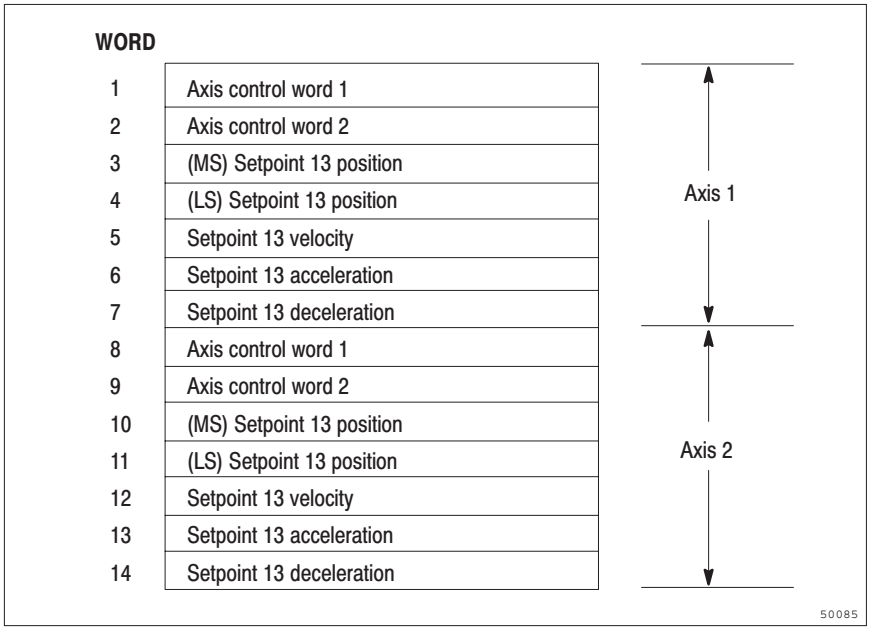


Figure E.2
Axis Control Word 1 (words 1 and 8)

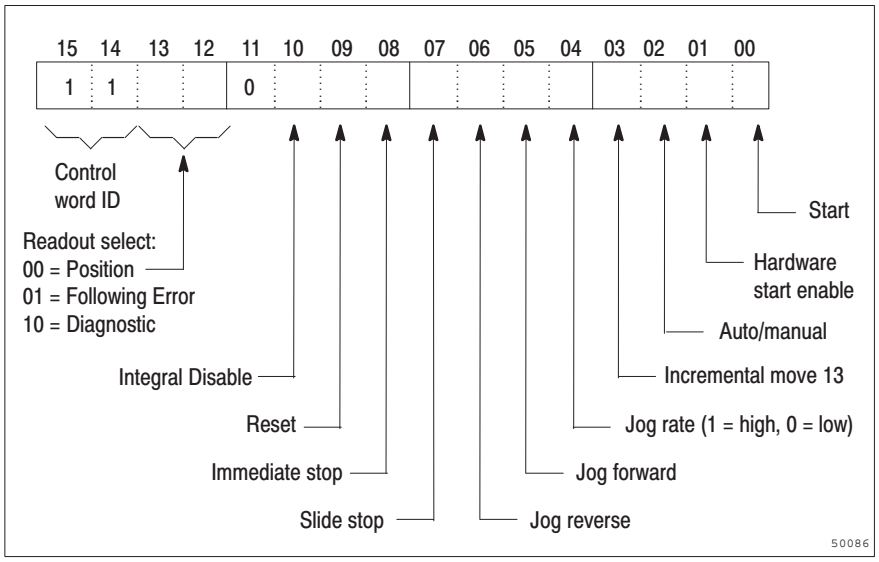


Figure E.3
Axis Control Word 2 (words 2 and 9)

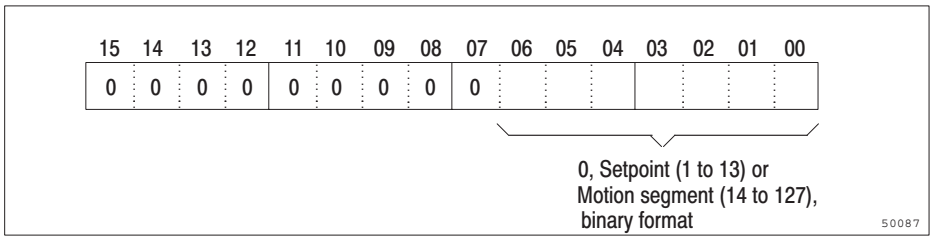


Figure E.4
Setpoint 13
Position Words (words 3, 4 and 10, 11)

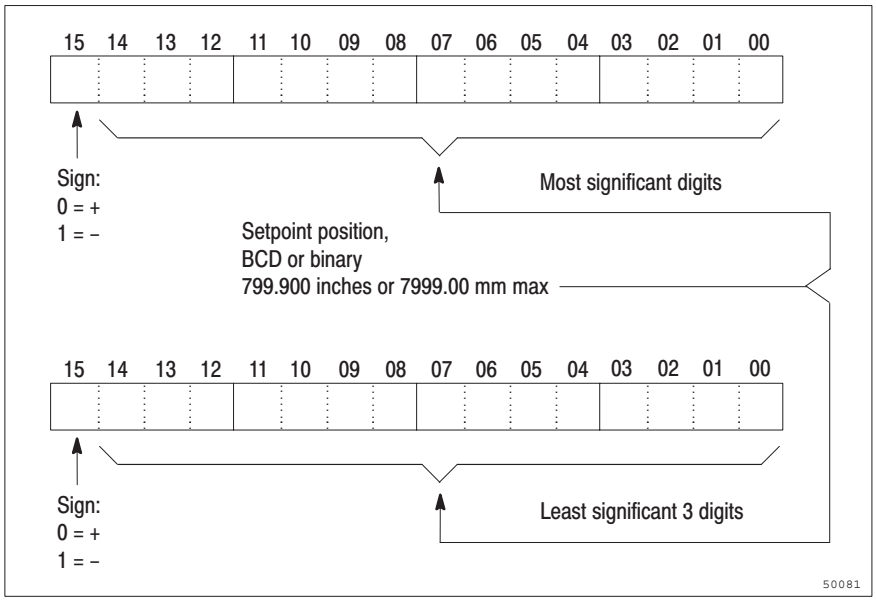

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Setpoint position,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50081

Figure E.5
Setpoint 13
Local Velocity Words (words 5 and 12)

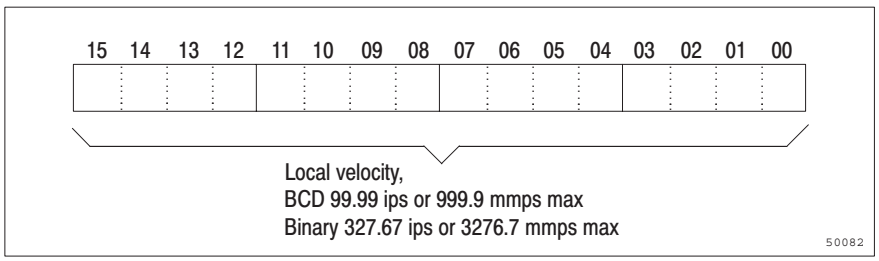
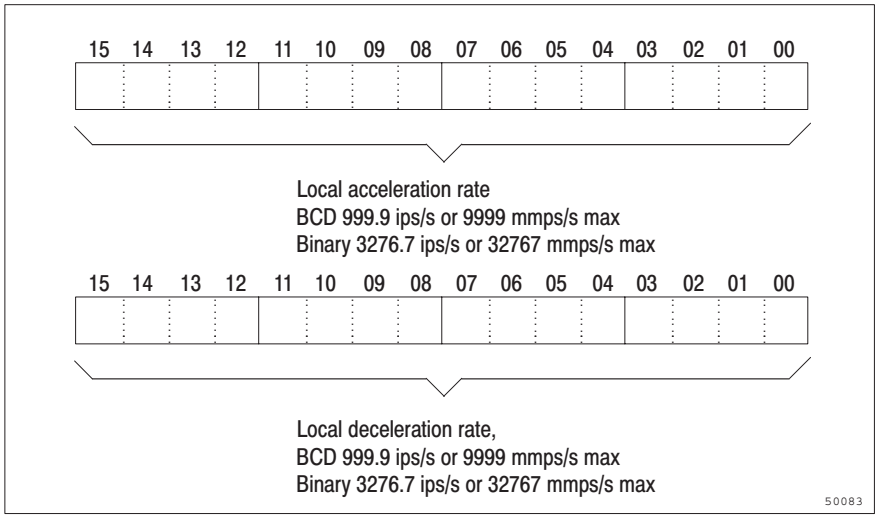


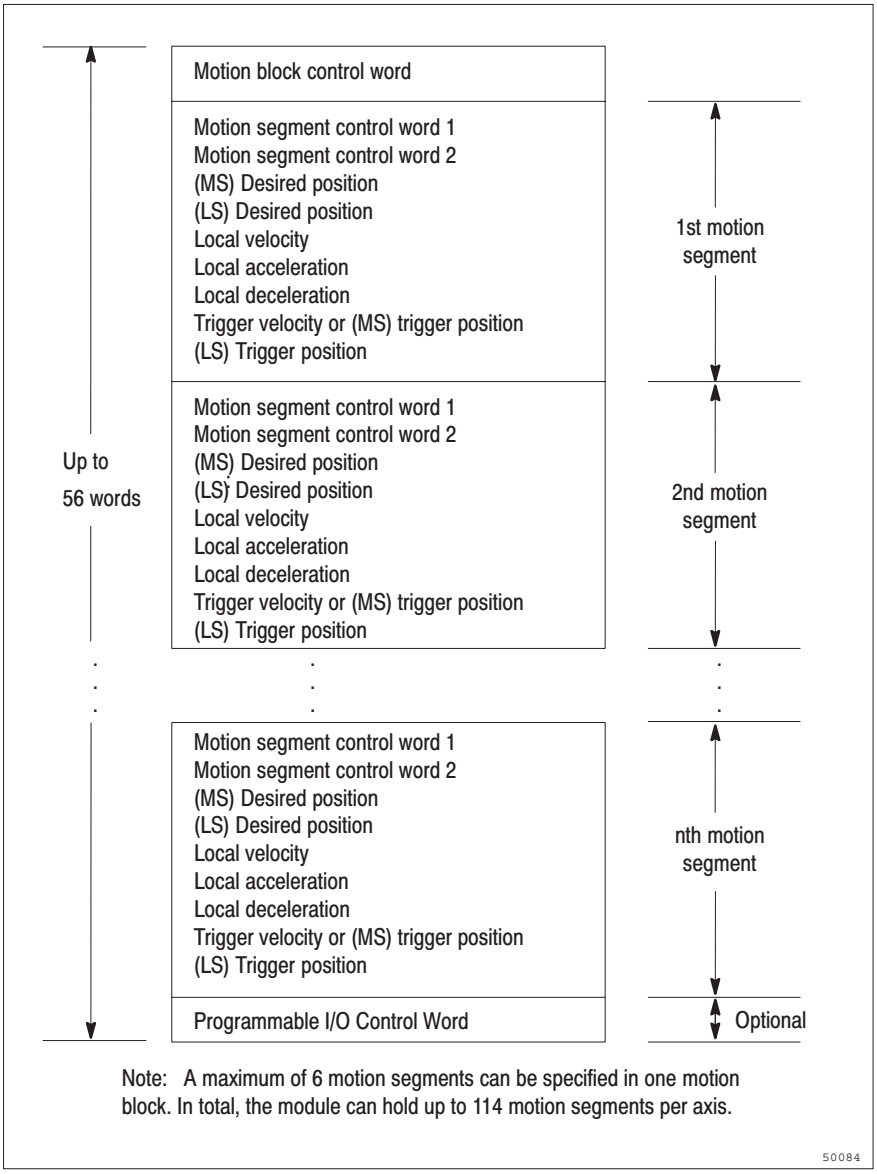
Figure E.6
Setpoint 13
Local Acceleration/Deceleration Words (words 6, 7 and 13, 14)


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local acceleration rate
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local deceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50083

Appendix F

Motion Block

Figure F.1
Motion Block Word Assignments



Note: A maximum of 6 motion segments can be specified in one motion block. In total, the module can hold up to 114 motion segments per axis.

Figure F.2
Motion Block Control Word

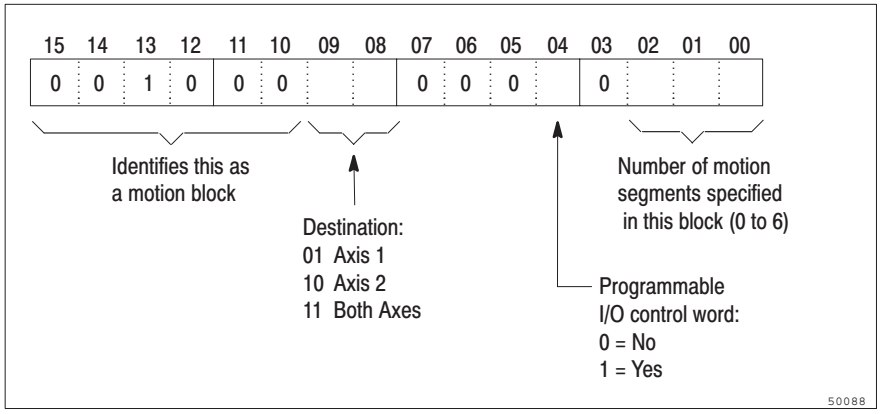

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0 0 1 0 0 0 0 0 0 0
Identifies this as a motion block
Destination:
01 Axis 1
10 Axis 2
11 Both Axes
Number of motion segments specified in this block (0 to 6)
Programmable I/O control word:
0 = No
1 = Yes
50088

Figure F.3
Programmable I/O Control Word

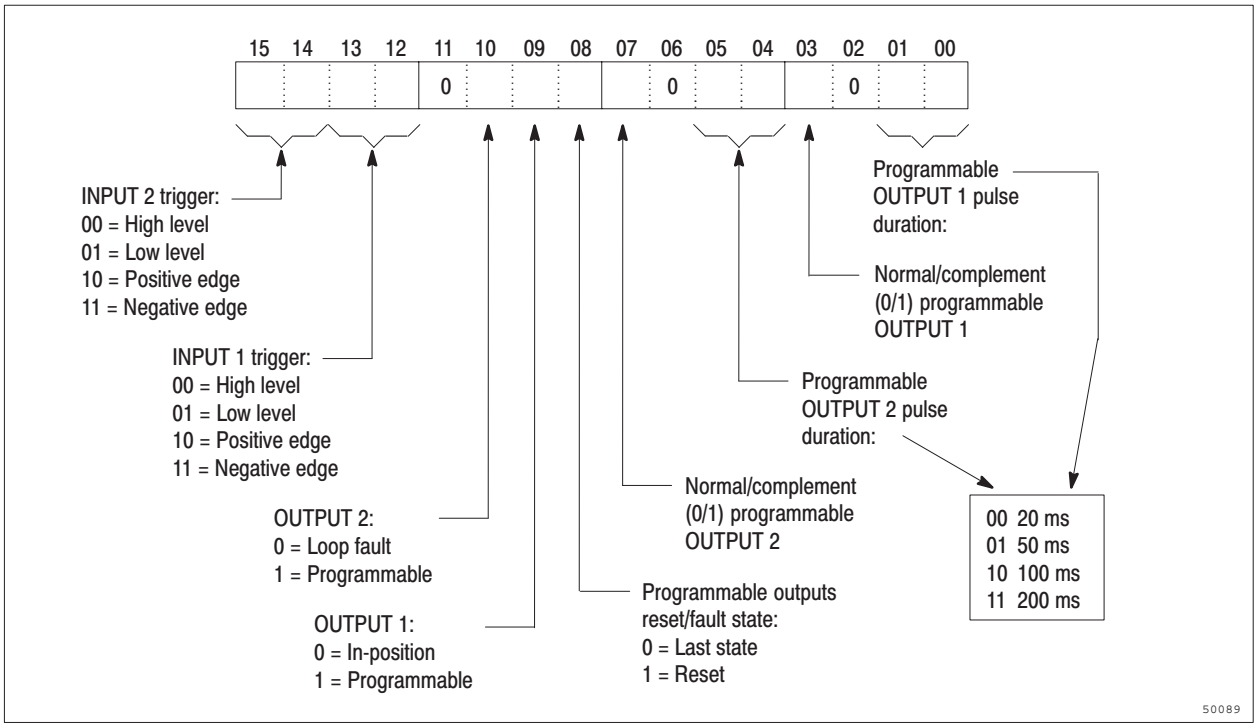


Figure F.4
Motion Segment Control Words

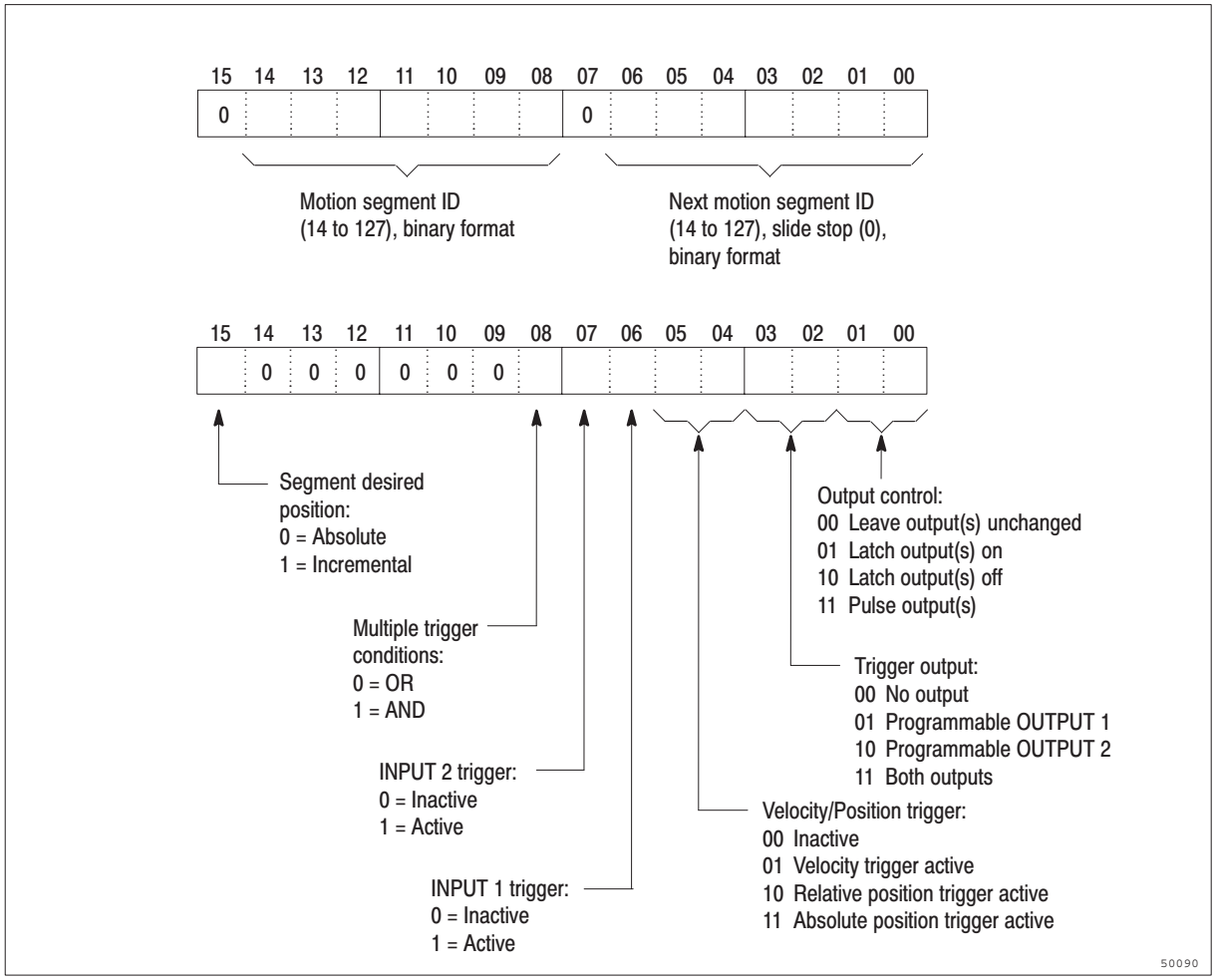


Figure F.5
Desired/Trigger Position Words

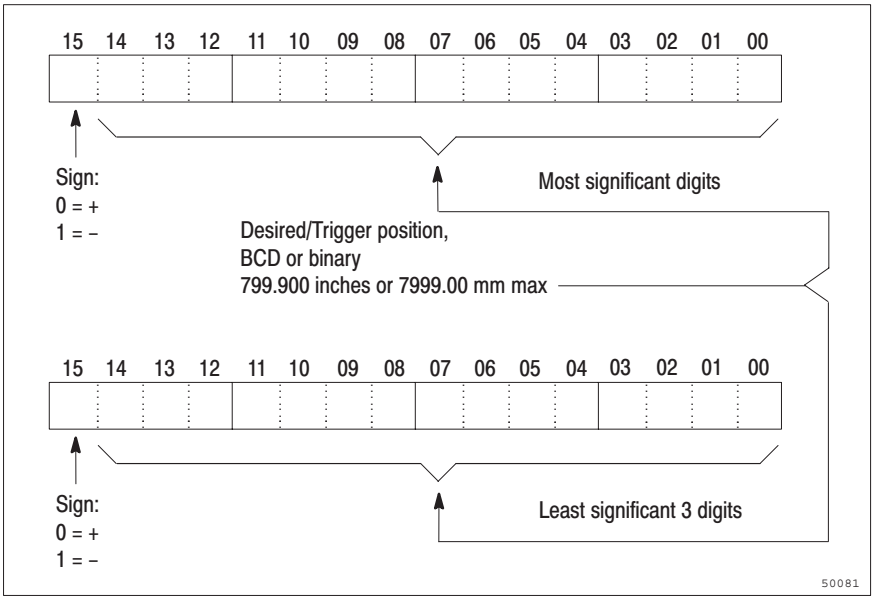

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Desired/Trigger position,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50081

Figure F.6
Local/Trigger Velocity Words

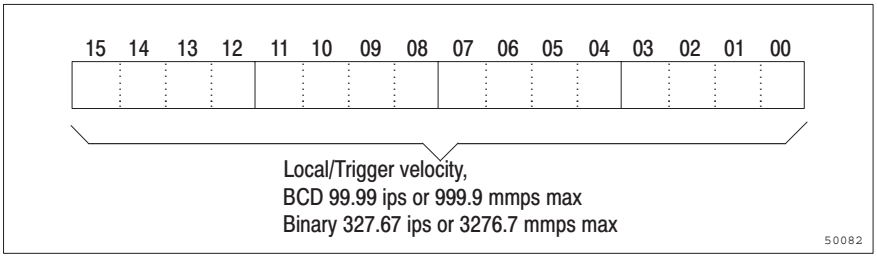
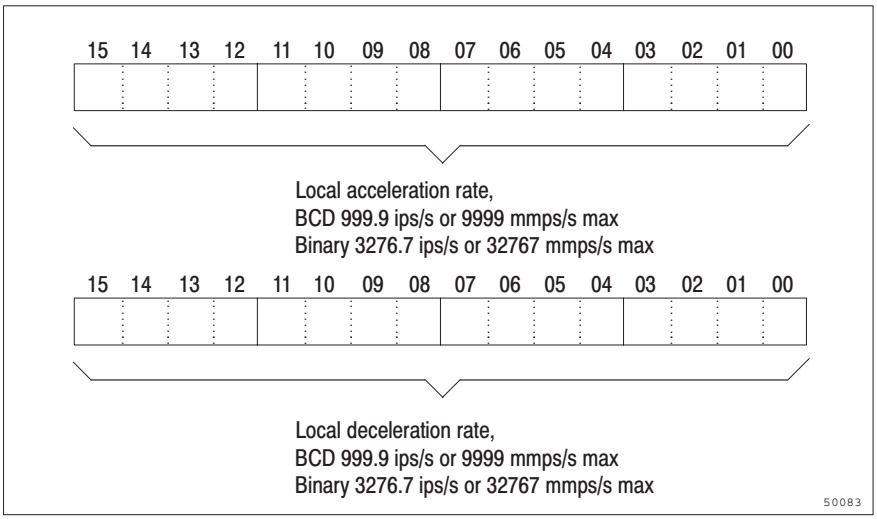


Figure F.7
Local Acceleration/Deceleration Words


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local acceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Local deceleration rate,
BCD 999.9 ips/s or 9999 mmps/s max
Binary 3276.7 ips/s or 32767 mmps/s max
50083

Appendix G

Hexadecimal Data Table Forms

For your convenience, we have included data table forms for each type of block, and both axes, where applicable, on the following pages. Copy these forms and fill it in with hexadecimal values for the parameter, setpoint, motion and command blocks, and necessary sequencer data for your PLC programs.

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 1 Block Description: Parameter

Data Table Address	Position	File Data				Description
N45:1	**1**	4	1			Parameter control word
2	**2**					Analog range
3	**3**					+ Analog calibration constant
4	**4**					– Analog calibration constant
5	**5**					(MS) Transducer calibration constant
6	**6**					(LS) Transducer calibration constant
7	**7**					(MS) Zero-position offset
8	**8**					(LS) Zero-position offset
9	**9**					+ Software travel limit
10	**10**					– Software travel limit
11	**11**					In-position band
12	**12**					PID band
13	**13**					Deadband
14	**14**					Excess following error
15	**15**					Maximum PID error
16	**16**	0				Integral term limit
17	**17**					Proportional gain
18	**18**					Gain break speed
19	**19**					Gain factor
20	**20**					Integral gain
21	**21**					Derivative gain
22	**22**	0				Feedforward gain
23	**23**					Global velocity
24	**24**					Global acceleration
25	**25**					Global deceleration
26	**26**					Velocity smoothing constant
27	**27**					Low jog rate
28	**28**					High jog rate
29	**29**	0	0	0	0	Reserved
30	**30**	0	0	0	0	Reserved

50102

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 2 Block Description: Parameter

Data Table Address	Position	File Data				Description
N45:31	**31**					Analog range
32	**32**					+ Analog calibration constant
33	**33**					– Analog calibration constant
34	**34**					(MS) Transducer calibration constant
35	**35**					(LS) Transducer calibration constant
36	**36**					(MS) Zero-position offset
37	**37**					(LS) Zero-position offset
38	**38**					+ Software travel limit
39	**39**					– Software travel limit
40	**40**					In-position band
41	**41**					PID band
42	**42**					Deadband
43	**43**					Excess following error
44	**44**					Maximum PID error
45	**45**	0				Integral term limit
46	**46**					Proportional gain
47	**47**					Gain break speed
48	**48**					Gain factor
49	**49**					Integral gain
50	**50**					Derivative gain
51	**51**	0				Feedforward gain
52	**52**					Global velocity
53	**53**					Global acceleration
54	**54**					Global deceleration
55	**55**					Velocity smoothing constant
56	**56**					Low jog rate
57	**57**					High jog rate
58	**58**	0	0	0	0	Reserved
59	**59**	0	0	0	0	Reserved

50103

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 1 Block Description: Setpoint

Data Table Address	Position	File Data				Description
N45:61	**1**	8		0		Setpoint block control word
62	**2**					Incremental/absolute word
63	**3**					Move #1 (MS) Setpoint position
64	**4**					(LS) Setpoint position
65	**5**					Local velocity
66	**6**					Local acceleration
67	**7**					Local deceleration
68	**8**					Move #2 (MS) Setpoint position
69	**9**					(LS) Setpoint position
70	**10**					Local velocity
71	**11**					Local acceleration
72	**12**					Local deceleration
73	**13**					Move #3 (MS) Setpoint position
74	**14**					(LS) Setpoint position
75	**15**					Local velocity
76	**16**					Local acceleration
77	**17**					Local deceleration
78	**18**					Move #4 (MS) Setpoint position
79	**19**					(LS) Setpoint position
80	**20**					Local velocity
81	**21**					Local acceleration
82	**22**					Local deceleration
83	**23**					Move #5 (MS) Setpoint position
84	**24**					(LS) Setpoint position
85	**25**					Local velocity
86	**26**					Local acceleration
87	**27**					Local deceleration
88	**28**					Move #6 (MS) Setpoint position
89	**29**					(LS) Setpoint position
90	**30**					Local velocity
91	**31**					Local acceleration
92	**32**					Local deceleration

50105

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 1 Block Description: Setpoint

Data Table Address	Position	File Data				Description
93	**33**					Move #7 (MS) Setpoint position
94	**34**					(LS) Setpoint position
95	**35**					Local velocity
96	**36**					Local acceleration
97	**37**					Local deceleration
98	**38**					Move #8 (MS) Setpoint position
99	**39**					(LS) Setpoint position
100	**40**					Local velocity
101	**41**					Local acceleration
102	**42**					Local deceleration
103	**43**					Move #9 (MS) Setpoint position
104	**44**					(LS) Setpoint position
105	**45**					Local velocity
106	**46**					Local acceleration
107	**47**					Local deceleration
108	**48**					Move #10 (MS) Setpoint position
109	**49**					(LS) Setpoint position
110	**50**					Local velocity
111	**51**					Local acceleration
112	**52**					Local deceleration
113	**53**					Move #11 (MS) Setpoint position
114	**54**					(LS) Setpoint position
115	**55**					Local velocity
116	**56**					Local acceleration
117	**57**					Local deceleration
118	**58**					Move #12 (MS) Setpoint position
119	**59**					(LS) Setpoint position
120	**60**					Local velocity
121	**61**					Local acceleration
122	**62**					Local deceleration

50106

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 2 Block Description: Setpoint

Data Table Address	Position	File Data				Description
N45:161	**1**	8		0		Setpoint block control word
162	**2**					Incremental/absolute word
163	**3**					Move #1 (MS) Setpoint position
164	**4**					(LS) Setpoint position
165	**5**					Local velocity
166	**6**					Local acceleration
167	**7**					Local deceleration
168	**8**					Move #2 (MS) Setpoint position
169	**9**					(LS) Setpoint position
170	**10**					Local velocity
171	**11**					Local acceleration
172	**12**					Local deceleration
173	**13**					Move #3 (MS) Setpoint position
174	**14**					(LS) Setpoint position
175	**15**					Local velocity
176	**16**					Local acceleration
177	**17**					Local deceleration
178	**18**					Move #4 (MS) Setpoint position
179	**19**					(LS) Setpoint position
180	**20**					Local velocity
181	**21**					Local acceleration
182	**22**					Local deceleration
183	**23**					Move #5 (MS) Setpoint position
184	**24**					(LS) Setpoint position
185	**25**					Local velocity
186	**26**					Local acceleration
187	**27**					Local deceleration
188	**28**					Move #6 (MS) Setpoint position
189	**29**					(LS) Setpoint position
190	**30**					Local velocity
191	**31**					Local acceleration
192	**32**					Local deceleration

50107

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 2 Block Description: Setpoint

Data Table Address	Position	File Data				Description
193	**33**					Move #7 (MS) Setpoint position
194	**34**					(LS) Setpoint position
195	**35**					Local velocity
196	**36**					Local acceleration
197	**37**					Local deceleration
198	**38**					Move #8 (MS) Setpoint position
199	**39**					(LS) Setpoint position
200	**40**					Local velocity
201	**41**					Local acceleration
202	**42**					Local deceleration
203	**43**					Move #9 (MS) Setpoint position
204	**44**					(LS) Setpoint position
205	**45**					Local velocity
206	**46**					Local acceleration
207	**47**					Local deceleration
208	**48**					Move #10 (MS) Setpoint position
209	**49**					(LS) Setpoint position
210	**50**					Local velocity
211	**51**					Local acceleration
212	**52**					Local deceleration
213	**53**					Move #11 (MS) Setpoint position
214	**54**					(LS) Setpoint position
215	**55**					Local velocity
216	**56**					Local acceleration
217	**57**					Local deceleration
218	**58**					Move #12 (MS) Setpoint position
219	**59**					(LS) Setpoint position
220	**60**					Local velocity
221	**61**					Local acceleration
222	**62**					Local deceleration

50108

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. _____ Block Description: Motion Block

Data Table Address	Position	File Data				Description
	1	2				Motion block control word
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	10					
	11					
	12					
	13					
	14					
	15					
	16					
	17					
	18					
	19					
	20					
	21					
	22					
	23					
	24					
	25					
	26					
	27					
	28					
	29					
	30					
	31					
	32					
	33					
	34					

50101

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. _____ Block Description: Motion Block

Data Table Address	Position	File Data				Description
	35					
	36					
	37					
	38					
	39					
	40					
	41					
	42					
	43					
	44					
	45					
	46					
	47					
	48					
	49					
	50					
	51					
	52					
	53					
	54					
	55					
	56					

50101

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. 1 and 2 Block Description: Command

Data Table Address	Position	File Data				Description	
N45:131	**1**					Axis 1	Axis control word 1
132	**2**						Axis control word 2
133	**3**						(MS) Setpoint 13 position
134	**4**						(LS) Setpoint 13 position
135	**5**						Setpoint 13 velocity
136	**6**						Setpoint 13 acceleration
137	**7**						Setpoint 13 deceleration
138	**8**					Axis 2	Axis control word 1
139	**9**						Axis control word 2
140	**10**						(MS) Setpoint 13 position
141	**11**						(LS) Setpoint 13 position
142	**12**						Setpoint 13 velocity
143	**13**						Setpoint 13 acceleration
144	**14**						Setpoint 13 deceleration

50109

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. _____ Block Description: Sequencer Data

Data Table Address	Position	File Data				Description
D9:0	1					
1	2					
2	3					
3	4					
4	5					
5	6					
6	7					
7	8					
8	9					
9	10					
10	11					
11	12					
12	13					
13	14					
14	15					
15	16					
16	17					
17	18					
18	19					
19	20					
20	21					
21	22					
22	23					
23	24					
24	25					
25	26					
26	27					
27	28					
28	29					
29	30					
30	31					
31	32					
32	33					
33	34					
34	35					

50104

Appendix G
Hexadecimal Data Table Forms

Project Name: _____ Page____ of ____

Designer: _____ Address ____ of ____

Date:_____ Axis No. _____ Block Description:_____

Data Table Address	Position	File Data				Description
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	10					
	11					
	12					
	13					
	14					
	15					
	16					
	17					
	18					
	19					
	20					
	21					
	22					
	23					
	24					
	25					
	26					
	27					
	28					
	29					
	30					
	31					
	32					
	33					
	34					
	35					

50104

Appendix H

Data Formats

Bit 3 in the parameter control word (word 1 in the parameter block) determines the format of the data contained in block transfer reads and writes. BCD format provides compatibility with older programmable controllers. Binary format provides compatibility with the PLC-5, which uses integer (16-bit 2's complement) data. This appendix explains both these numbering formats.

BCD

BCD (Binary Coded Decimal) is a numbering format by which decimal digits are directly represented by 4-bit groups of binary numbers. Here's how the decimal digits 0 to 9 are represented:

Table H.A
Binary Representation

Decimal Digit	Binary Representation
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Example: To represent the number 8761 (decimal) using BCD notation, set the 16 bits of the word like this:

1000	0111	0110	0001
8	7	6	1

Important: In BCD format, none of the four-bit groupings may contain a value greater than 1001 (binary).

2's Complement Binary

To complement a number means to change it to a negative number. 2's complement binary is a method used to represent signed integers in binary notation. This method is used with PLC-3 and PLC-5 processors.

Following are two methods to get the negative of a number using the 2's complement method.

Bit Inversion Method

To get the 2's complement of a number using the bit inversion method you must invert each bit from right to left after the first 1.

Example: To represent -1524 (decimal) in 16-bit 2's complement format, we start with the binary equivalent of the positive of the number.

1524 (decimal) = 0000 0101 1111 0100 (Binary)

Next, we invert all the bits of the number from right to left starting with the bit after the first 1 that we encounter.

-1524 (decimal) = 1111 1010 0000 1100

Subtraction Method

To get the 16-bit 2's complement of a number using the subtraction method you must subtract the number from 2^{16}.

Thus to represent -1524 (decimal):

```
 1 0000 0000 0000 0000
-0000 0101 1111 0100
---------------------
 1111 1010 0000 1100
```

Implied Decimal

Many of the words in the module's data blocks use implied decimal points. This means that, although the value you want to enter into a word may have a fractional component, you only enter the digits, not the decimal point, when you type it into the programmable controller data table.

Example

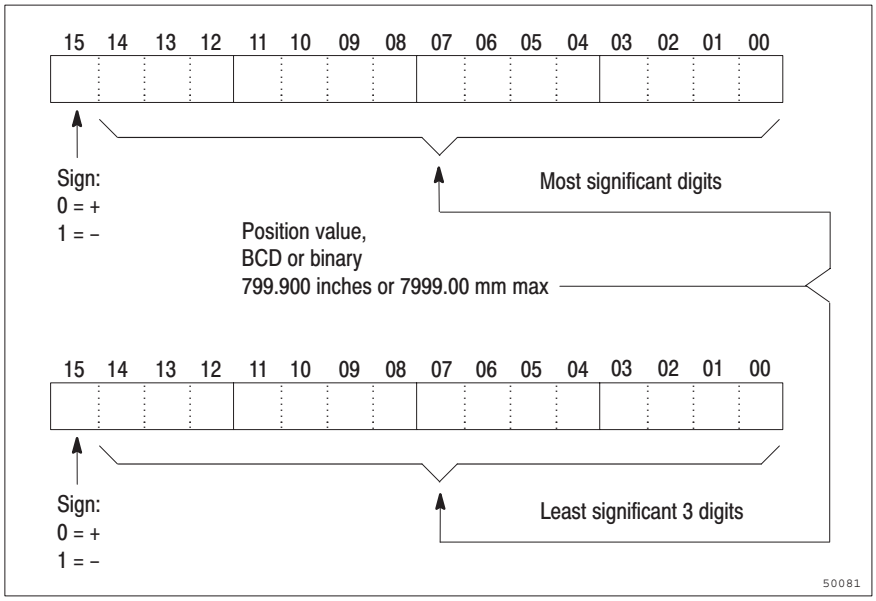
You want to program a global velocity of 1.50 inches/second for axis 1. This value has an implied decimal between the digits 1 and 5. The decimal point is implied because you don't actually type it when you enter the value into the programmable controller data table. Instead, you enter 150. When the programmable controller writes this value into Word 23 of the module's parameter block, the module assumes the decimal point and interprets the value correctly as 1.50 ips. Notice that you must enter the two least significant digits to the right of the implied decimal point, including zeros.

Important: You must enter the correct number of digits to the right of the implied decimal place. You may have to round your desired value or fill it to the right with zeros. Check the format of each word to determine where the implied decimal point is.

Position Format

The setpoint position, positioning error, and zero-position offset all use a double word format to represent positive or negative values. This format is illustrated in Figure H.1.

Figure H.1
Position Format


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Most significant digits
Position value,
BCD or binary
799.900 inches or 7999.00 mm max
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Sign:
0 = +
1 = –
Least significant 3 digits
50081

A sign bit is placed in each word to allow negative binary numbers even with the first word zeroed. Simply signing the first word in this case would not work in binary mode because a word with a value of zero and the sign bit on (i.e., a negative zero) is not equal to zero in the 16-bit 2's complement system.

If the first word of a negative number is zero, turn on the sign bit in the second word. For negative numbers that use both words, the use of the sign bit in the second word is optional (see example).

Table H.B gives several examples of how you would enter BCD and binary positions.

Table H.B
Binary and BCD Positions

	Binary Format		BCD Format	
Position	**1st Word**	**2nd Word**	**1st Word**	**2nd Word**
+60.000 in	60	0	0060	0000
+1524.00 mm	152	400	0152	0400
+0.999 in	0	999	0000	0999
+9.99 mm	0	999	0000	0999
-60.000 in	-60	0	8060	0000 or 8000
-1524.00 mm	-152	400 or -400	8152	0400 or 8400
-0.500 in	0	-500	0000 or 8000	8500

Double Word Position Format

The Linear Positioning Module supports the full range of values from -32,768 to 32,767 in the second word of the setpoint position, positioning error and zero-position offset parameters, when using binary format.

This flexibility means that you can often enter the same parameter value in different ways. You must still use two words in binary format if your numbers fall outside the 32,767 to -32,768 range.

For example, to enter a setpoint of 31.999 you could either:

- enter the value 31 in the first word and 999 in the second

or

- leave the first word zeroed and enter the value 31,999 in the second word

Both methods result in a setpoint value of 31.999.

Product Specifications

Location
- 1771 Universal I/O chassis
- One slot

Sampling Period
- 2 milliseconds for both loops (i.e., both axis positions are read simultaneously every 2 milliseconds)

Highest Velocity
- 327.67 inches per second
- 3276.7 millimeters per second

Switchable Analog Output Range
- -10 VDC to +10 VDC @ up to 10 mA
- -20 mA to +20 mA (up to 600 ohms)
- -50 mA to +50 mA (up to 240 ohms)
- -100 mA to +100 mA (up to 120 ohms)

Analog Output Resolution
- 12-bit resolution

Transducer Interface
- 180 inches/4572 millimeters maximum stroke length
- Resolution: 0.002 inches (one circulation)
- RS422 Drivers/Receivers
- External interrogation: 3 microseconds pulse width
- Single gate pulse, 1.680 milliseconds max - 117 MHz internal counters

Transducer Compatibility
- Manufacturer: MTS Systems Corporation
  Temposonics II™ with digital personality module
  Power requirements: +15 VDC @ 150 mA, -15 VDC @ 100 mA
  Temposonics I Transducer with digital interface box
  Power requirements: ±15 VDC, 5 VDC ±5%

- Manufacturer: Balluff Inc.
  BTL Linear Displacement Transducer
  Power requirements: 24 VDC ±10% @150 mA

- Manufacturer: Santest Co. Ltd.
  Magnetostrictive Displacement Transducer: Model GYRG
  Power requirements: ±15 VDC ±10% @ 48 mA

- Manufacturer: Lucas Schaevitz
  Magnerule Plus: Model MRU-XXX-700
  Power requirements: ±15 VDC ±4 VDC @ 80 mA

Discrete Inputs
- Logic 0: 0 to 4 VDC
- Logic 1: 10 to 30 VDC
- Input Current:
  8 mA @ 12 VDC
  16 mA @ 24 VDC

Discrete Outputs
- Single ended source
- Logic 0: No voltage supplied to output (OFF state)
- Logic 1: User-supplied voltage applied to output (ON state)
- Current:
  Maximum source drive 100 mA (ON state)
  Maximum source leakage 1.0 mA (OFF state)

Backplane Power Requirements
- 1.6 A maximum (1.1 A typical) @ +5 VDC

Transducer Interface External Power Requirements
- +5 VDC ±5% @ 300 mA maximum

Analog Interface External Power Requirements
- +15 VDC ±5% @ 540 mA maximum
- –15 VDC ±5% @ 360 mA maximum

Discrete Input External Power Requirements
- +15 VDC minimum
- +24 VDC maximum @ 50 mA maximum

Discrete Output External Power Requirements
- +30 VDC maximum @ 400 mA maximum
- Max. voltage drop from supply to output = 1.6 VDC @ 100 mA (+11.6 VDC min. is required for compatibility with discrete inputs)

Thermal Dissipation
- 12.0 watts typical (89 BTU/hour)
- 18.0 watts maximum (120 BTU/hour)

Keying
- Between 16 and 18, 30 and 32

Wiring Arm
- 40 terminal 1771–WN
- 14 gauge stranded wire (max)
- 3/64ths inch insulation (max)
- Category 2

Electrical Isolation
- 1500V rms (transient) isolation is achieved by optoelectronic coupling between I/O circuits and control logic

Environmental Conditions
- Operating temperature: 0°C to 60°C (32°F to 140°F)
- Storage temperature: –40°C to 85°C (–40°F to 185°F)
- Relative humidity: 5% to 95% non-condensing

Index

A

Absolute Positioning, 7-29

Acceleration, 7-34
- Global, 7-24
- Local, 7-32
- With Velocity Smoothing, 7-24

Analog Calibration Constants, 7-6

Analog Fault Bit, 6-9

Analog Outputs, 4-7
- Interface Terminals, 4-3

Analog Range, 7-5

Auto Mode, 3-4, 7-35

Auto Mode Bit, 6-5

Auto/Manual Bit, 6-5, 7-35

Auto/Manual Input, 4-6

Axis Control Words, 7-32

Axis Motion, Concepts, 7-19

Axis Polarity, 7-9

B

Binary/BCD Bit, 7-4

Blended Moves, 9-1

Block Transfer Instructions, 6-1

C

Circulations, 2-2

Command Block, 3-2, 9-11

Control Word ID Bits, 7-38

D

Data Block, 7-1
- Command, 7-1
- Motion, 9-1
- Parameter, 7-1
- Setpoint, 7-1

Data Blocks, 6-1

Deadband, 2-7, 7-15

Deceleration, 7-34
- Global, 7-24
- Local, 7-32

Derivative Control, 2-6

Derivative Gain, 2-6, 7-22

Desired Velocity, 2-4

Diagnostic Valid Bit, 6-8

Diagnostic Words, 6-9
- Error Codes, 6-10

Discrete Inputs
- Disabling, 6-5
- Enabling, 6-5
- Fault, 6-8
- Power Supply, 5-14
- Priority Over Jog Bits, 7-36
- Terminals, 4-2
- Voltage and Current Requirements, 5-12

Discrete Outputs, 5-13
- Characteristics, 4-8

Done Bit, 6-4

E

Electrostatic Discharge, Avoiding, 5-2

Error Valid Bit, 6-7

Error Words, 6-9

Excess Following Error, 7-16

Excess Following Error Bit, 6-8

F

Feedback Fault Bit, 6-9

Feedforwarding, 2-5

Following Error, 2-4, 7-16
- Excess Following Error Bit, 6-8

G

Gain
- Derivative, 2-6, 7-22
- Feedforward, 2-5
- Integral, 2-6, 7-21
- Proportional, 2-4, 7-18

Gain Break Speed, 7-19

Gain Factor, 7-19

General Purpose Inputs, 4-7, 5-16

H

Hardware Start Enable Bit, 7-35

Hardware Start Input, 4-6

Hardware Stop Input, 4-7

I

Immediate Stop Bit, 6-8, 7-37
In-Position Band, 7-13
In-Position Bit, 6-4
Inch/Metric Bit, 7-3
Incremental Move 13 Bit, 7-35
Incremental Positioning, 7-29, 7-34
Incremental/Absolute Word, 7-29
INPUT 2, Trigger, 9-10
Input Bits, 6-6
Input Triggers, 9-7, 9-10
Inputs Enabled Bit, 6-5
Installation, 5-1
Integral Control, 2-5
Integral Gain, 2-6, 7-21
Integral Limit Reached Bit, 6-8
Integral Term Limit, 7-17
Internal Fault Bit, 6-9
INTPUT 1, Trigger, 9-10

J

Jog Bits, 6-6
Jog Rate Select Bit, 7-35
Jog Rates, 3-5, 7-26, 7-35
Jogging, 3-5

L

Linear Positioning Module
  Description, 1-1
  In a Positioning System, 1-4
Loop Fault Output, 6-8, 6-9

M

Manual Mode, 3-5, 7-35
Maximum PID Error, 7-16
Module
  Inserting in I/O Chassis, 5-5
  Installation, 5-1
  Location, 5-1
Motion Block, 9-1, 9-2, 9-12
  Control Word, 9-4
Motion Segment, 9-1, 9-2
  Control Word, 9-8, 9-9
  Desired Position Word, 9-11
  Example, 9-3
  Local Acceleration Word, 9-11
  Local Deceleration Word, 9-11
  Local Velocity Word, 9-11

N

Next Setpoint Bits, 7-38
Number-of-Axes Bit, 7-3

O

OUTPUT 1, 4-9, 5-21, 6-4, 9-5, 9-7, 9-9
OUTPUT 2, 4-9, 5-21, 6-8, 9-5, 9-6, 9-7, 9-9

P

Parameter Block, 3-2
  Control Word, 7-3
PID Active Bit, 6-5
PID Band, 2-7, 7-14
PID Error Bit, 6-8
PID Error, Maximum, 7-16
Position Valid Bit, 6-8
Position Words, 6-9, 6-11
Positioning
  Absolute, 7-29
  Incremental, 7-29, 7-34
Power Supplies
  Avoiding Backplane Power Supply Overload, 5-1
  Connecting, 5-10, 5-11, 5-14, 5-19, 5-21
Programmable I/O, 9-1, 9-5, 9-7
  Control Word, 9-5
  Default Configuration, 9-8
Programming Error Bit, 6-5
Proportional Gain, 2-4, 7-18

R

Rate Control, 2-6
Read Operations, 3-2
Readout Select Bit, 7-38

Ready Bit, 6-3
Reset Bit, 7-37
Reset Control, 2-5

S

Setpoint 13 Words, 7-39
Setpoint Block, 3-2
  Control Word, 7-28
Setpoint Moves, 3-4
Setpoint Number, 6-7
Setpoint Position, 7-30
Setpoints Received Bit, 6-4
Shielded Cables, 5-6
Slide Stop Bit, 7-36
Software Travel Limits, 7-9
Start Bit, 6-6, 7-34
Status Block, 9-11
Status Words, 6-3
Stop Bit, 6-6

T

Transducer, 1-2
  Connecting, 5-12
  Interface Terminals, 4-2
  Power Supply, 5-11
Transducer Calibration Procedure, 8-7
Trigger Condition, 9-1
Trigger Conditions, 9-3, 9-7
  Multiple, 9-10

V

Velocity Curve Smoothing, 3-3
Velocity Smoothing Constant, 7-24
Velocity, Desired, 2-4
Velocity/Position Trigger, 9-10

W

Write Operations, 3-2

Z

Zero-Position Offset, 7-8

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Allen-Bradley has been helping its customers improve productivity and quality for 90 years. A-B designs, manufactures and supports a broad range of control and automation products worldwide. They include logic processors, power and motion control devices, man-machine interfaces and sensors. Allen-Bradley is a subsidiary of Rockwell International, one of the world's leading technology companies.

With major offices worldwide.

Algeria • Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • New Zealand • Norway • Oman • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Switzerland • Taiwan • Thailand • The Netherlands • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

World Headquarters, Allen-Bradley, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1771-6.5.44, September 1993
Supersedes 1771-6.5.44, August 1988

PN 95592501
Copyright 1993 Allen-Bradley and Dynapro Systems Inc. Printed in Canada